

Mihaela BALCĂ, Georgeta CRĂCIUNESCU, Andrei FLOREA
Oana-Maria LUPAȘCU, Carmen MINCĂ (coord.)
Cristiana POPESCU, Octavian-Lucian POPA, Silviu SĂCUIU

INFORMATICĂ



CULEGERE DE PROBLEME

Rezolvate în limbajele Python₃ / C++

PENTRU LICEU

ISBN: 978-630-6559-22-0
2025 - Editura L&S Soft | Infobits Academy
<https://ebooks.infobits.ro>

Copyright 2025 © Editura L&S Soft Management Impex S.R.L.

Toate drepturile asupra acestei lucrări aparțin editurii L&S SOFT.

Reproducerea integrală sau parțială a textului din această carte este posibilă doar cu acordul în scris al editurii L&S SOFT.

ISBN: 978-630-6559-22-0

**MATERIALUL ESTE DISTRIBUIT GRATUIT.
SPOR LA STUDIU!**

Infobits Academy

Editura L&S SOFT

*Pentru comenzi, folosiți următoarele
date de contact:*

Telefon: 0727.731.947

E-mail: hello@infobits.ro

www.infobits.ro

<https://ebooks.infobits.ro/>



Compania noastră oferă de peste 30 de ani **manuale școlare aprobate de Ministerul Educației și auxiliare ce respectă programa școlară, aplicații online**, precum și **cursuri de Informatică și T.I.C.**, utile oricărei persoane care dorește să se pregătească în aceste domenii.

Autorii și colaboratorii noștri sunt oameni dedicați procesului educațional, cu o **experiență pedagogică vastă** și autori de manuale școlare și auxiliare avizate de Ministerul Educației.

Cuprins

I.	Operatori, expresii și tipuri de date	4
II.	Structura liniară.....	12
III.	Structura alternativă	27
IV.	Structuri repetitive	56
	IV.1. Instrucțiunea repetitivă while	56
	IV.2. Instrucțiunea repetitivă for	79
V.	Șiruri de caractere	110
VI.	Liste	141
VII.	Fișiere text	189
VIII.	Răspunsuri	233
IX.	Bibliografie	234

I. Operatori, expresii și tipuri de date

1. Pentru fiecare dintre itemii următori alegeți litera corespunzătoare răspunsului corect.

1) Indicați ce se va afișa pe ecran în urma executării comenzii Python : `print("10+23")`

- a) 33 b) 10+23=33 c) 10+23 d) EROARE

2) Indicați ce se va afișa pe ecran în urma executării comenzii Python: `print(1001//37)`

- a) 27 b) 27.05405 c) 0.2705e1 d) 27.05

3) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
Z = "LIMBAJUL" "PYTHON"
print(Z)
```

- a. LIMBAJUL c. LIMBAJUL PYTHON
b. PYTHON d. LIMBAJULPYTHON

4) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
x='AbC'
print(x+' DeEF')
```

- a. AbC b. DeEF c. AbCDeEF d. AbC DeEF

5) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
# comentariu: atribuire
var = 2026 # continut variabila
# cerinta
s="# Ce valoare avem?"
print(s)
```

- a. # comentariu: atribuire c. #Cevaloareavem?
b. 2026 d. # Ce valoare avem?

6) Indicați ce se va afișa pe ecran în urma executării instrucțiunii Python următoare: `print(13.999, int(13.999))`

- a. 13.99 13 b. 13.999 13 c. 13.999 13.999 d. 13.999 14

7) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
Zi = 'Luni'
Zi = 20.07
Zi = 2026
print(Zi)
```

- a. Luni b. 20.07 c. 2026 d. 20.07 2026

8) Indicați ce se va afișa pe ecran în urma executării comenzii Python : `print(23%7)`

- a. 2 b. 3 c. 3.2857 d. 23%7

9) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
x, y, z = 15, 2.34, "Test!"
print(x,y,z)
```

- a. 15 15 15 b. 15 2 34 c. 15 2.34 "Test!" d. 15 2.34 Test!

10) Știind că variabila **x=True** și variabila **y=False**, indicați care dintre expresiile Python următoare are valoarea **False**.

- a. **x and y** b. **x or y** c. **not not x** d. **x and y or x**

11) Se consideră secvența de comenzi Python alăturată.

Ce se va afișa pe ecran în urma executării comenzilor, în ordinea din secvență?

```
sir = "Programare"
x=10
y=20
list = [10, 20, 30, 40, 50]
print('A' not in sir)
print('g' in sir)
print('r' in sir)
print("gram" in sir)
print(x in list)
print(y not in list)
```

- | | | | |
|---------|---------|---------|---------|
| a. True | b. True | c. True | d. True |
| True | False | True | False |
| True | True | True | False |
| True | True | False | True |
| True | True | True | True |
| False | False | False | False |

12) Secvența de comenzi Python alăturată este utilizată pentru inițializarea variabilelor **x, y, z** și **w**.

Ce valori au aceste variabile după executarea secvenței?

```
x = (True == 1)
y = (False == 0)
z = 2 + True
w = False - 10
```

- | | | | |
|-------------------|-------------------|-------------------|----------------|
| a. x= True | b. x= True | c. x= True | d. x= 1 |
| y= True | y= True | y= True | y= 1 |
| z= 3 | z= 3 | z= True | z= 3 |
| w= False | w= -10 | w= False | w= -10 |

13) Fie expresia logică: **(x % y == 0) or not (y % x != 0)**. Ce valori pot avea variabilele **x** și **y**, astfel încât în urma evaluării expresiei, expresia să aibă valoarea **True**?

- | | | | |
|----------------|----------------|-----------------|-----------------|
| a. x= 4 | b. x= 5 | c. x= 14 | d. x= 15 |
| y= 9 | y= 11 | y= 28 | y= 14 |

14) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
lista = [1, 2, 3, 4, 5]
x = 3 in lista
y = 20 not in lista
print(x,y)
```

- a. **True True** b. **True False** c. **False False** d. **False True**

- 15) Ce valori au variabilele **E1** și **E2** la finalul executării secvenței de comenzi Python alăturate?

```
x,y = 10, 10
L1 = [1, 2, 3, 4]
L2 = [1, 2, 3, 4]
E1 = (id(x) == id(y))
E2 = (id(L1) == id(L2))
```

- a. **E1=True**
E2=True b. **E1=True**
E2=False c. **E1=False**
E2=False d. **E1=False**
E2=True

2. Se consideră secvența de comenzi Python alăturată.
Știind că se citesc valorile 3 și 6 pentru variabilele **x** și **y**,

```
x= input()
y= int(input())
z= int(x) + y
```

În această ordine, asociați fiecărei comenzi Python din prima coloană a tabelului următor, valoarea/tipul afișat/ă pe ecran, ca urmare a executării, precizat în a doua coloană.

```
print(type(x))
print(type(y))
print(x+str(y))
print(type(z))
print(z)
print(type(2026))
print(type(12.05))
print(type(-7.1e3))
print(type('x'))
```

```
36
<class 'int'>
<class 'str'>
<class 'float'>
9
```

3. Evaluați expresiile Python de mai jos precizând rezultatul și tipul expresiilor.

- a) `4*x+9//3`, unde variabila **x** are valoarea 2
b) `7*x-5/2`, unde variabila **x** are valoarea 5
c) `x>2*3`, unde variabila **x** are valoarea 7
d) `(2*x-1)%5`, unde variabila **x** are valoarea 6
e) `3*(x+4)-2`, unde variabila **x** are valoare 2.0
f) `True+False+10`

4. Asociați fiecărei constante Python din prima linie categoria din care face parte:

`"xyz"` `'156'` `142e-2` `2026` `723.45` `0xB12` `'x'` `0.182` `0o34` `2.25e1` `"523.26"`

Constante numerice
întregi

Constante numerice
reale

Constante șir de
caractere

5. Știind că variabilele **x**, **y** și **z** au valorile **x=24**, **y=16**, **z=4**, asociați fiecărei expresii Python din prima coloană a tabelului următor, valoarea corespunzătoare din a doua coloană.

a) <code>24 & 16</code>	1) <code>-17</code>
b) <code>24 16</code>	2) <code>1</code>
c) <code>24 ^ 16</code>	3) <code>8</code>
d) <code>~ 16</code>	4) <code>16</code>
e) <code>24 << 4</code>	5) <code>24</code>
f) <code>16 >> 4</code>	6) <code>384</code>

6. Evaluați expresiile Python de mai jos precizând rezultatul și tipul expresiilor.

a) `"abc"+"12"` b) `12+int("3")` c) `3*"abc"` d) `'b' in "abcd"`

7. Pentru fiecare dintre itemii următori alegeți litera corespunzătoare răspunsului corect.

- 1) Fie codul Python de mai jos.

```
a= 17
b= 31.2026
c='Python'
d= 17+3j
print(a, " ", b, " ", c, " ", d)
```

Indicați care este tipul variabilei d:

a) `complex` b) `întreg` c) `real` d) `șir de caractere`

- 2) Indicați ce se va afișa pe ecran în urma executării secvenței de comenzi Python de mai jos:

```
a= 10+31j
b= 17+3j
print(a+b)
```

a) `27+34i` b) `27+34j` c) `a+b` d) `Eroare`

- 3) Indicați rezultatul cărei sume este afișat pe ecran în urma executării secvenței de comenzi Python alăturate.

```
n=int(input("n= "))
print(n*(n+1))
```

a. `1+3+5+...+(2*n-1)` c. `1+2+3+...+n`
b. `2+6+10+...+(4*n-2)` d. `2+4+6+...+2*n`

- 4) Ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate?

```
sir = [1, 3, 17, 31, 2000, 2026]
a=(sir[3]-(sir[0]+sir[1]+sir[2]))*200
print(a in sir)
print(sir)
```

- a. True
[1, 3, 17, 31, 2000, 2026]
- b. False
[1, 3, 17, 31, 2000, 2026]
- c. True
[1, 3, 17, 31, 2000, 2006]
- d. False
Eroare

- 5) Indicați rezultatul cărei sume este afișat pe ecran în urma executării secvenței de comenzi Python alăturate.

```
n=int(input("n= "))
print(n*(n+1)//2)
```

- a. $1+3+5+\dots+(2*n-1)$
- b. $2+6+10+\dots+(4*n-2)$
- c. $1+2+3+\dots+n$
- d. $2+4+6+\dots+2*n$

- 6) Indicați ce se va afișa pe ecran în urma executării secvenței de comenzi Python alăturate:

```
sir1 = [3, 17, 2000]
sir2 = [7, 19, 2026]
print(sir1+sir2)
```

- a. [3, 17, 2000, 7, 19, 2026]
- b. [3, 7, 17, 19, 2000, 2026]
- c. [3, 17, 2000]
- d. [7, 19, 2026]

- 7) Indicați rezultatul cărei sume este afișat pe ecran în urma executării secvenței de comenzi Python alăturate.

```
n=int(input("n= "))
print(n*n)
```

- a. $1+3+5+\dots+(2*n-1)$
- b. $2+6+10+\dots+(4*n-2)$
- c. $1+2+3+\dots+n$
- d. $2+4+6+\dots+2*n$

- 8) Indicați ce se va afișa pe ecran în urma executării secvenței de comenzi Python:

```
a=1234
```

```
print(a//1000+10*((a//100)%10)+100*((a//10)%10)+1000*(a%10))
```

- a. 3412 b. 3421 c. 3.4231 d. 4321

- 9) Indicați ce se va afișa pe ecran în urma executării secvenței de comenzi Python:

```
a=1234
```

```
print(a//1000%2+((a//100)%10)%2+((a//10)%10)%2+(a%10)%2)
```

- a. 1 b. 2 c. 3 d. 4

- 10) Știind că variabila $a=31$, $b=10$ și variabila $c=2002$, indicați care dintre expresiile Python următoare are valoarea **True**.

- a. $a < b$ or $b > c$
- b. $a == c$ or $b \% 2 == 1$
- c. $a + b == c$ or $b > c$
- d. $\text{not}(a + b == c)$

- 11) Se consideră secvența de comenzi Python alăturată. Ce se va afișa pe ecran în urma executării comenzilor, în ordinea din secvență?

```
text1 = "Eu iubesc"
text2 = " sa invatat"
a=31
b=17
sir = [1, 3, 17, 31, 2000, 2002, 2026]
print('E' in text1)
print("inv" in text2)
print(text1+text2)
print(a//b not in sir)
```

- | | |
|----------------------|----------------------|
| a. True | c. True |
| False | True |
| Eu iubesc sa invatat | Eu iubesc sa invatat |
| True | True |
| b. True | d. True |
| False | True |
| Eu iubesc sa invatat | Eu iubesc sa invatat |
| False | False |

- 12) Ce valori au aceste variabile după executarea secvenței de comenzi Python de mai jos?

```
a=[1,2,3,4,5,6,7]
print(a[4], " ", a[6]-a[1])
```

- | | | | |
|--------|--------|--------|--------|
| a. 5 5 | b. 5 6 | c. 4 5 | d. 4 6 |
|--------|--------|--------|--------|

8. Pentru fiecare dintre itemii următori alegeți litera corespunzătoare răspunsului corect.

- 1) Indicați ce se va afișa pe ecran în urma executării secvenței de comenzi Python de mai jos:

```
x = 3
y = 2
z = x ** y
print(z % 4)
```

- | | | | |
|------|------|------|------|
| a) 1 | b) 3 | c) 0 | d) 2 |
|------|------|------|------|

- 2) Indicați ce se va afișa pe ecran în urma executării secvenței de comenzi Python de mai jos:

```
a = 10
b = 3
print(a % b * b)
```

- | | | | |
|------|-------|------|------|
| a) 1 | b) 10 | c) 0 | d) 3 |
|------|-------|------|------|

- ```
x = True
y = False
print(not x or y and x)
```

c. None

d. False

- ```
a = 3
b = 5
a += b % 2
b //= a
print(a, b, bool(b))
```

c.3 1 True

```
d.5 0 False
```

- ```
a = 10
b = 3
print(a // b, a % b, a / b)
```

c. 3 1 3.3333333

d. 3.0 1.0 3.3333333

- b = 46**

d.37

- $$s_3 = s_1 + s_2$$

d. False

- d.  $a + b + c == 20$

- 9) Se consideră secvența de comenzi Python alăturată.  
Ce se va afișa pe ecran în urma executării comenzilor, în ordinea din secvență?

```
text = "Python e cool"
print("e" in text)
print("cool" in text)
print(text[7:9])
```

- |       |       |       |      |
|-------|-------|-------|------|
| a.    | b.    | c.    | d.   |
| False | False | True  | True |
| False | True  | False | True |
| e     | e     | e     | e    |

- 10) Ce valori se afisează după executarea secvenței de comenzi Python de mai jos?

```
a = [10, 20, 30, 40, 50]
x = a[2] + a[0]
y = a[4] - a[3]
print(x, y)
```

- |          |          |          |          |
|----------|----------|----------|----------|
| a. 40 10 | b. 30 20 | c. 40 20 | d. 50 10 |
|----------|----------|----------|----------|

## II. Structura liniară

1. Scrieți un program Python/C++ care să citească de la tastatură, de pe linii separate, două numere întregi **a** și **b**, apoi să se calculeze și să afișeze, pe ecran, suma numerelor citite. De exemplu, pentru **a=25** și **b=13**, va afișa: 38

| Limbajul Python                                                          | Limbajul C/C++                                                                                                                                                                                              |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a = int(input("a=")) b = int(input("b=")) print("suma=", a+b)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int a,b;     cout&lt;&lt;"a="; cin&gt;&gt;a;     cout&lt;&lt;"b="; cin&gt;&gt;b;     cout&lt;&lt;"a+b="&lt;&lt;a+b;     return 0;}</pre> |

2. Scrieți un program Python/C++ care să citească de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale **a** și **b**, apoi să calculeze și să afișeze, pe ecran, diferența, produsul și câtul lor, în această ordine. De exemplu, pentru **a=25** și **b=13**, se va afișa: 12 325 1

| Limbajul Python                                                                                              | Limbajul C/C++                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre># valorile se citesc de pe # aceeași linie a, b = map(int, input().split()) print(a-b, a*b, a//b)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int a,b;     cin&gt;&gt;a&gt;&gt;b;     cout&lt;&lt;a-b&lt;&lt;" "&lt;&lt;a*b;     cout&lt;&lt;" "&lt;&lt;a/b;     return 0; }</pre> |

3. Scrieți un program Python/C++ care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale **a** și **b**, apoi determină și afișează, pe ecran, ultima cifră a sumei celor două numere naturale. De exemplu, pentru **a=25** și **b=18**, se va afișa: 3.

| Limbajul Python                                                      | Limbajul C/C++                                                                                                                                         |
|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a,b=input().split() a=int(a) b=int(b) c=(a+b)%10 print(c)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int a,b;     cin&gt;&gt;a&gt;&gt;b;     cout&lt;&lt;(a+b)%10;     return 0; }</pre> |

4. Într-o clasă sunt **F** fete și **B** băieți. Fiecare fată rezolvă câte **X** probleme de fizică pe zi, iar fiecare băiat rezolvă **Y** probleme pe zi. Câte probleme rezolvă elevii din clasă în **Z** zile? Scrieți un program Python/C++ care să rezolve această problemă.

De exemplu, pentru **F=7, B=5, X=3, Y=2, Z=5** se va afișa 155.

| Limbajul Python                                                   | Limbajul C/C++                                                                                                                                                                               |
|-------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>F,B,X,Y,Z=map(int, input().split()) print((X*F+Y*B)*Z)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int F,B,X,Y,Z;     cin&gt;&gt;F&gt;&gt;B&gt;&gt;X&gt;&gt;Y&gt;&gt;Z;     cout&lt;&lt;(X*F+Y*B)*Z;     return 0; }</pre> |

5. La un club de pictură participă **n** copii. Ei trebuie să picteze, câte doi, un tablou. Știind că fiecare copil va picta câte un tablou cu fiecare copil din club, câte tablouri vor fi pictate în total? Scrieți un program Python/C++ care să citească de la tastatură numărul natural **n** și care să afișeze numărul total de tablouri care vor fi pictate de copii. De exemplu, pentru **n=4**, se va afișa valoarea 6.

| Limbajul Python                             | Limbajul C/C++                                                                                                                                 |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) print(n*(n-1)//2)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n;     cin&gt;&gt;n;     cout&lt;&lt;n*(n-1)/2;     return 0; }</pre> |

6. Scrieți un program Python/C++ care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale **a** și **b**, apoi formează un nou număr prin alipirea cifrei unităților lui **a** cu cifra unităților lui **b**. Programul afișează pe ecran numărul format. De exemplu, pentru **a=123** și **b=45**, programul va afișa valoarea 35.

| Limbajul Python                                                            | Limbajul C/C++                                                                                                                                                            |
|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a,b=input().split() a=int(a) b=int(b) c=(a%10)*10+b%10 print(c)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int a,b,c;     cin&gt;&gt;a&gt;&gt;b;     c=(a%10)*10+b%10;     cout&lt;&lt;c;     return 0; }</pre> |

7. Scrieți un program Python/C++ care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale **a** și **b** (**a>10**). Programul construiește un nou număr prin alipirea ultimelor două cifre ale lui **a** la finalul **b**, apoi afișează pe ecran numărul format. De exemplu, pentru **a=123** și **b=45**, programul va afișa valoarea 4523.

| Limbajul Python                                                               | Limbajul C/C++                                                                                                                                                                |
|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a,b=input().split() a=int(a) b=int(b) c=a%100 b=b*100+c print(b) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a,b,c;     cin&gt;&gt;a&gt;&gt;b;     c=a%100; b=b*100+c;     cout&lt;&lt;b;     return 0; } </pre> |

8. Fie  $n$  un număr natural, format din exact 4 cifre, toate nenule. Scrieți un program Python/C++ care citește numărul  $n$  și formează un nou număr prin alipirea ultimelor două cifre ale lui  $n$  cu primele două cifre ale sale. Programul afișează pe ecran numărul format. De exemplu, pentru  $n=1234$ , programul va afișa valoarea 3412.

| Limbajul Python                                                   | Limbajul C/C++                                                                                                                                                                |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) x=n%100 y=n//100 n=x*100 + y print(n) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,x,y;     cin&gt;&gt;n;     x=n%100; y=n/100;     n=x*100+y; cout&lt;&lt;n;     return 0; } </pre> |

9. Scrieți un program Python/C++ care citește de la tastatură o cifră  $c$  și un număr natural  $n$ , format din exact 4 cifre, toate nenule, apoi modifică numărul  $n$  înlocuindu-i cifra unităților cu cifra  $c$ . Programul afișează pe ecran numărul modificat. De exemplu, pentru  $c=8$  și  $n=1234$ , programul va afișa valoarea 1238.

| Limbajul Python                                                     | Limbajul C/C++                                                                                                                                                   |
|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> c, n = map(int, input().split()) n=n//10*10+c print(n) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,c;     cin&gt;&gt;c&gt;&gt;n; n=n/10*10+c;     cout&lt;&lt;n;     return 0; } </pre> |

10. Fie  $n$  un număr natural  $n$ , format din exact 4 cifre, toate nenule. Scrieți un program Python/C++ care citește numărul  $n$  și îl modifică mutând cifra miilor la finalul numărului citit. Programul afișează pe ecran numărul modificat. De exemplu, pentru  $n=1234$ , programul va afișa valoarea 2341.

| Limbajul Python                                                   | Limbajul C/C++                                                                                                                                                                 |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n =int(input()) x=n%1000 y=n//1000; n=x*10+y; print(n)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int n,x,y;     cin&gt;&gt;n;     x=n%1000; y=n/1000;     n=x*10+y;     cout&lt;&lt;n;     return 0; }</pre> |

11. Scrieți un program Python/C++ care să citească de la tastatură, de pe linii separate, două numere întregi **a** și **b**, apoi să se calculeze și să afișeze, pe ecran, câtul și restul numerelor citite. De exemplu, pentru **a=31** și **b=17**, va afișa: 1 14

| Limbajul Python                                                                                | Limbajul C/C++                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a = int(input("a=")) b = int(input("b=")) print("catul=",a//b) print("restul=",a%b)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int a,b;     cout&lt;&lt;"a="; cin&gt;&gt;a;     cout&lt;&lt;"b="; cin&gt;&gt;b;     cout&lt;&lt;"a div b="&lt;&lt;a/b;     cout&lt;&lt;"a mod b="&lt;&lt;a%b;     return 0; }</pre> |

12. Scrieți un program Python/C++ care să citească de la tastatură, un număr natural **a** cu exact trei cifre, apoi să calculeze și să afișeze, pe ecran, numărul cifrelor pare și numărul cifrelor impare, în această ordine. De exemplu, pentru **a=123**, se va afișa: 1 2

| Limbajul Python                                                                            | Limbajul C/C++                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a=int(input("a= ")) nri=a//100%2+a//10%10%2+a%10%2 nri=3-nri print(nri," ",nri)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int a,nrp,nri;     cout&lt;&lt;"a= ";     cin&gt;&gt;a;     nri=a/100%2+a/10%10%2+     nri=nri+a%10%2     nrp=3-nri;     cout&lt;&lt;nrp&lt;&lt;" "&lt;&lt;nri;     return 0; }</pre> |

13. Scrieți un program Python/C++ care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale cu exact două cifre **a** și **b**, apoi determină și afișează, pe ecran, numărul format din ultima cifră a lui **a** și prima cifră a lui **b**. De exemplu, pentru **a=25** și **b=81**, se va afișa: 58.

| Limbajul Python                                                               | Limbajul C/C++                                                                                                                                                                                                |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a = int(input("a= ")) b = int(input("b= ")) print((a%10)*10+b//10)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int a,b;     cout&lt;&lt;"a= "; cin&gt;&gt;a;     cout&lt;&lt;"b= "; cin&gt;&gt;b;     cout&lt;&lt;(a%10)*10+b/10;     return 0; }</pre> |

14. Scrieți un program care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale cu exact două cifre **a** și **b**, apoi determină și afișează, pe ecran, numărul format prin alipirea la numărul **a**, numărul **b**, spațiu și invers. De exemplu, pentru **a=25** și **b=81**, se va afișa: 2581 8125,

| Limbajul Python                                                                   | Limbajul C/C++                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a = int(input("a= ")) b = int(input("b= ")) print(a*100+b," ",b*100+a)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {int a,b;     cout&lt;&lt;"a= ";     cin&gt;&gt;a;     cout&lt;&lt;"b= ";     cin&gt;&gt;b;     cout&lt;&lt;a*100+b&lt;&lt;" ";&lt;&lt;b*100+a;     return 0; }</pre> |

15. La un club de fotbal participă **a** băieți talentați. Ei nu încep antrenamentul până nu își dau mâna între ei o singură dată. Scrieți un program care afișează câte strângeri de mână au loc. De exemplu, pentru **a=10**, se va afișa valoarea 45.

| Limbajul Python                                    | Limbajul C/C++                                                                                                                                 |
|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a = int(input("a= ")) print(a*(a-1)//2)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int a;     cin&gt;&gt;a;     cout&lt;&lt;a*(a-1)/2;     return 0; }</pre> |

16. Un vas plin cu apă cântărește  $x$  grame. După ce se introduce în vas o bilă de  $p$  grame, vasul cântărește  $y$  grame.

Scrieți un program **Python/C++** care să citească valorile reale  $x$ ,  $p$ ,  $y$  și care să determine densitatea materialului din care este confecționată bila.

**De exemplu**, pentru  $x=50$ ,  $p=12$  și  $y=60.5$  programul va afișa  $8 \text{ (g/cm}^2\text{)}$

| Limbajul Python                                                                   | Limbajul C/C++                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>X,P,Y=map(float, input().split()) densitate=P/(X+P-Y) print(densitate)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     float X,P,Y,densitate;     cin&gt;&gt;X&gt;&gt;P&gt;&gt;Y;     densitate=P/(X+P-Y);     cout&lt;&lt;densitate;     return 0; }</pre> |

17. Scrieți un program **Python** care citește de la tastatură un număr real  $L$  ( $n < 10^4$ ) reprezentând lungimea laturii unui triunghi echilateral și afișează, pe rânduri diferite, cu două zecimale exacte, perimetrul triunghiului, aria triunghiului, raza cercului înscris în triunghi, raza cercului circumscris triunghiului și lungimea înălțimii, ca în exemplu.

**Exemplu:** pentru  $L=3$  atunci se afișează:

$P=9.00$

$A=3.90$

$r=0.87$

$R=1.73$

$h=2.60$

| Limbajul Python                                                                                                                                                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import math L = float(input()) P = 3 * L h = (L * math.sqrt(3)) / 2 A = (L ** 2 * math.sqrt(3)) / 4 r = h / 3 R = L / math.sqrt(3) print(f"P={P:.2f}") print(f"A={A:.2f}") print(f"r={r:.2f}") print(f"R={R:.2f}") print(f"h={h:.2f}")</pre> | <pre>#include &lt;iostream&gt; #include &lt;cmath&gt; #include &lt;iomanip&gt; using namespace std; int main() {     double L;     cin&gt;&gt;L;     double P=3*L;     double h=(L*sqrt(3))/2;     double A=(L*L*sqrt(3))/4;     double r=h/3;     double R=L/sqrt(3);     cout&lt;&lt;fixed&lt;&lt;setprecision(2);     cout&lt;&lt;"P="&lt;&lt;P&lt;&lt;"\n";     cout&lt;&lt;"A="&lt;&lt;A&lt;&lt;"\n";     cout&lt;&lt;"r="&lt;&lt;r&lt;&lt;"\n";     cout&lt;&lt;"R="&lt;&lt;R&lt;&lt;"\n";     cout&lt;&lt;"h="&lt;&lt;h&lt;&lt;"\n";     return 0; }</pre> |

18. Scrieți un program în **Python/C++**, care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale nenule  $x$  și  $y$  ( $x < 10$  și  $y < 10$ ), apoi afișează, pe ecran, separate printr-un spațiu,  $x^y$  și  $y^x$ , în această ordine. De exemplu, pentru  $x=5$  și  $y=2$ , se va afișa:  $25 \ 32$

| Limbajul Python                                               | Limbajul C/C++                                                                                                                                                                                             |
|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x, y = map(int, input().split()) print(x**y, y**x)</pre> | <pre>#include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std; int main() { unsigned int x,y;   cin&gt;&gt;x&gt;&gt;y;   cout&lt;&lt;pow(x,y)&lt;&lt;" "&lt;&lt;pow(y,x) ;   return 0; }</pre> |

19. Scrieți un program în Python/C++, care citește de la tastatură un număr natural nenul  $n$ , cu cel mult patru cifre, apoi calculează și afișează, pe ecran, în câte cifre de zero se termină produsul  $n!$  (unde  $n! = 1 \cdot 2 \cdot \dots \cdot n$ ). De exemplu, pentru  $n=32$ , se va afișa: 7

| Limbajul Python                                                                      | Limbajul C/C++                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input("n= ")) nrzero=n//5+n//25+n//125+n//625+n//3125 print(nrzero)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int n,nrzero;   cout&lt;&lt;"n= ";   cin&gt;&gt;n;   nrzero=n/5+n/25+n/125;   nrzero+=(n/625+n/3125);   cout&lt;&lt;nrzero;   return 0; }</pre> |

20. Scrieți un program în Python/C++, care citește de la tastatură un număr natural nenul  $n$ , cu cel mult trei cifre, apoi calculează și afișează, pe ecran, valoarea expresiei  $1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + n \cdot (n+1)$ . De exemplu, pentru  $n=3$ , se va afișa: 20

| Limbajul Python                                            | Limbajul C/C++                                                                                                                                                                   |
|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input("n= ")) E=n*(n+1)*(n+2)//3 print(E)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { unsigned int n,E;   cout&lt;&lt;"n= "; cin&gt;&gt;n;   E= n*(n+1)*(n+2)/3;   cout&lt;&lt;E;   return 0; }</pre> |

21. Scrieți un program în Python/C++, care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, trei numere naturale distincte,  $a$ ,  $b$  și  $c$ , reprezentând cifre prime, apoi determină și afișează, pe ecran, cifra primă lipsă. De exemplu, pentru  $a=5$ ,  $b=7$  și  $c=2$ , se va afișa: 3

| Limbajul Python                                                         | Limbajul C/C++                                                                                                                                                                      |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a, b, c = map(int, input().split()) d=a^b^c^2^3^5^7 print(d)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { unsigned int a, b, c, d;   cin&gt;&gt;a&gt;&gt;b&gt;&gt;c;   d=a^b^c^2^3^5^7;   cout&lt;&lt;d;   return 0; }</pre> |

22. Inițial, un antreprenor deține  $n$  acțiuni la o companie din domeniul IT. În  $m$  zile consecutive, antreprenorul își dublează numărul acțiunilor față de cât deținea în ziua anterioară, astfel: după prima zi are  $2*n$  acțiuni, după a doua zi are  $4*n$  acțiuni etc. Scrieți un program în Python/C++, care citește de la tastatură un număr natural nenul  $n$ , cu cel mult două cifre, reprezentând numărul inițial de acțiuni, apoi citește de la tastatură un număr natural nenul  $m$ ,  $m \leq 11$ . Programul afișează, pe ecran, câte acțiuni deține antreprenorul la finalul zilei a  $m$ -a.

**De exemplu**, pentru  $n=90$ ,  $m=5$  se va afișa valoarea 2880

| Limbajul Python                                                             | Limbajul C/C++                                                                                                                                                                                              |
|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input("n= ")) m = int(input("m= "))  print(n &lt;&lt; m)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int n,m;      cout&lt;&lt;"n= "; cin&gt;&gt;n;     cout&lt;&lt;"m= "; cin&gt;&gt;m;     cout&lt;&lt;(n &lt;&lt; m);     return 0;}</pre> |

23. Scrieți un program **Python/C++** care citește de la tastatură un număr  $n$ , de două cifre, și construiește un nou număr **rez** prin alipirea la sfârșitul lui  $n$  a sumei cifrelor lui. Programul va afișa pe ecran numărul construit. **De exemplu**, pentru  $n=47$  programul va afișa **rez = 4711**, deoarece suma cifrelor este  $4 + 7 = 11$ .

| Limbajul Python                                                                                                                     | Limbajul C/C++                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input()) zeci = n // 10 unitati = n % 10 suma = zeci + unitati rez = int(str(n) + str(suma)) print("rez =", rez)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { int n,zeci,unitati,suma,rez;   cin &gt;&gt; n;   zeci=n/10; unitati=n%10;   suma = zeci + unitati;   n= n*((suma&gt;9) ? 100: 10);   rez = n + suma;   cout &lt;&lt; "rez = " &lt;&lt; rez;   return 0; }</pre> |

24. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$ , de trei cifre, toate cifrele fiind diferite de 0, apoi construiește în oglindit acestuia și îl afișează pe ecran. **De exemplu**, pentru  $n = 372$ , se va afișa:  $m = 273$ .

| Limbajul Python                                                                                                                      | Limbajul C/C++                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) sute=n//100 zeci=(n//10)%10 unitati=n%10 # construim oglinditul m=unitati*100+zeci*10+sute print("m =", m)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { int n, m;   cin &gt;&gt; n;   int sute=n/100, zeci=(n/10)%10;   int unitati=n%10;   // construim oglinditul   m=unitati*100+zeci*10+sute;   cout &lt;&lt; "m = " &lt;&lt; m;   return 0; }</pre> |

25. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$ , format din două cifre. Programul va forma un nou număr **rez**, obținut prin dublarea fiecărei cifre a numărului  $n$ , în ordinea în care apar, și va afișa acest nou număr pe ecran. De exemplu, dacă se citește  $n = 75$ , atunci cifrele 7 și 5 devin 77 și 55, iar rezultatul afișat va fi 7755.

| Limbajul Python                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input())  zec = n // 10 unit = n % 10  rez = int(str(zec) * 2 + str(unit) * 2) print("rez =", rez)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int n;   cin &gt;&gt; n;   int zec = n / 10;   int unit = n % 10;   int rez = zec * 1000;   rez=rez+zec*100+unit*10+unit;   cout &lt;&lt; "rez = " &lt;&lt; rez;   return 0; }</pre> |

26. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$ , format din exact șase cifre. Programul va separa numărul în două grupuri de câte trei cifre (primele trei și ultimele trei), va calcula suma cifrelor pentru fiecare grup și va afișa diferența absolută dintre cele două sume.  
Exemplu: Dacă se citește  $n = 321457$ , atunci vom avea grupurile : 321 și 457  
→  $|6 - 16| = 10$

| Limbajul Python                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = input() # Grupa 1: primele 3 cifre g1 = int(n[0]) + int(n[1]) + int(n[2]) # Grupa 2: ultimele 3 cifre g2 = int(n[3]) + int(n[4]) + int(n[5]) # Diferența absolută diferenta = abs(g1 - g2) print("Diferența este:", diferenta)</pre> |

## Limbajul C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
 int n;
 cin >> n;
 int c6 = n % 10; n /= 10;
 int c5 = n % 10; n /= 10;
 int c4 = n % 10; n /= 10;
 int c3 = n % 10; n /= 10;
 int c2 = n % 10; n /= 10;
 int c1 = n % 10;
 int suma1 = c1 + c2 + c3;
 int suma2 = c4 + c5 + c6;
 int diferenta = abs(suma1 - suma2);
 cout << "Diferenta este: " << diferenta;
 return 0;
}
```

27. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$ , format din exact patru cifre. Programul va:

- înmulți fiecare cifră a numărului cu poziția ei în număr (de la stânga la dreapta),
- concatenând valorile obținute pentru a forma un nou număr rez,
- și va afișa acest rezultat.
- Exemplu: Dacă  $n = 1234$ , atunci, rezultatul este 14916

| Limbajul Python                                                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = input() rez = "" rez += str(int(n[0]) * 1) rez += str(int(n[1]) * 2) rez += str(int(n[2]) * 3) rez += str(int(n[3]) * 4)  print("rez =", rez)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     // extragem fiecare cifră     int c4 = n % 10; n /= 10;     int c3 = n % 10; n /= 10;     int c2 = n % 10; n /= 10;     int c1 = n;     int p1 = c1 * 1;     int p2 = c2 * 2;     int p3 = c3 * 3;     int p4 = c4 * 4;     // concatenăm valorile     cout &lt;&lt; "rez = " &lt;&lt; p1 &lt;&lt; p2;     cout &lt;&lt; p3 &lt;&lt; p4;     return 0; }</pre> |

28. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$ , format din exact cinci cifre. Programul va:

- extrage a doua și a patra cifră a numărului,
- calculează modulul diferenței dintre cele două cifre,
- plasează cifra rezultată între prima și ultima cifră ale numărului, în această ordine,
- și afișează rezultatul sub forma unui nou număr de 3 cifre.

Exemplu: Dacă  $n = 45618$ , atunci noul număr va fi 448

| Limbajul Python                                                                                                                                                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) # Extragem fiecare cifră c5 = n % 10 c4 = (n // 10) % 10 c3 = (n // 100) % 10 c2 = (n // 1000) % 10 c1 = n // 10000 diff = abs(c2 - c4) rez = c1 * 100 + diff * 10 + c5 print("rez =", rez) </pre> | <pre> #include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     // Extragem fiecare cifră     int c5 = n % 10;     int c4 = (n / 10) % 10;     int c3 = (n / 100) % 10;     int c2 = (n / 1000) % 10;     int c1 = n / 10000;     int diff = abs(c2 - c4);     int rez = c1*100+diff* 10 +c5;     cout &lt;&lt; "rez = " &lt;&lt; rez;     return 0; } </pre> |

29. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $d$ , format din exact patru cifre, reprezentând o distanță galactică. Programul va calcula suma celor patru cifre ale numărului, va înmulți această sumă cu prima cifră a numărului și va afișa rezultatul, reprezentând estimarea consumului de combustibil.

Exemplu: Dacă  $d = 2431$ , atunci consumul estimat va fi 20.

| Limbajul Python                                                                                                                                                                                                                          | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> d = int(input()) # Extragem cifrele c4 = d % 10 c3 = (d // 10) % 10 c2 = (d // 100) % 10 c1 = d // 1000 # Calculăm suma și consumul suma = c1 + c2 + c3 + c4 consum = suma * c1 print("Estimarea consumului este:", consum) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int d;     cin &gt;&gt; d;     // Extragem cifrele     int c4 = d % 10;     int c3 = (d / 10) % 10;     int c2 = (d / 100) % 10;     int c1 = d / 1000;     // Calculăm suma și consumul     int suma = c1 + c2 + c3 + c4;     int consum = suma * c1;     cout&lt;&lt;"Estimarea consumului";     cout&lt;&lt;" este: " &lt;&lt; consum;     return 0; } </pre> |

30. Scrieți un program în **Python/C++**, care citește de la tastatură, de pe aceeași linie, separate printr-un spațiu, două numere naturale nenule  $a$  și  $b$  ( $a < 10$  și  $b < 10$ ), apoi afișează, pe ecran, separate printr-un spațiu,  $ab$  și  $ba$ , în această ordine.

De exemplu, pentru  $a=7$  și  $b=3$ , se va afișa: 73 37

| Limbajul Python                                                                                   | Limbajul C/C++                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a, b = map(int, input().split()) ab = a * 10 + b ba = b * 10 + a print(f"{ab} {ba}") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned int a,b;     cin&gt;&gt;a&gt;&gt;b;     cout&lt;&lt;a*10+b&lt;&lt;" "&lt;&lt;b*10+a;     return 0; } </pre> |

31. Scrieți un program în **Python/C++**, care citește de la tastatură două numere. Afișați suma dintre cifra unităților primului număr și cifra zecilor celui de al doilea.

**De exemplu**, pentru a = 123 și b = 456 se va afișa: 3+5=8

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a= int(input("Primul numar: ")) b= int(input("Al doilea numar: ")) # Extragem cifra unităților din primul număr cifra_unitati = a % 10 # Extragem cifra zecilor din al doilea număr cifra_zeci = (b // 10) % 10  # Calculăm și afișăm suma suma = cifra_unitati + cifra_zeci print("Suma cifrei unitatilor primului numar si a cifrei zecilor celui de-al doilea este:", suma) </pre>                                                                      |
| Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a, b;     cout &lt;&lt; "Primul numar: ";     cin &gt;&gt; a;     cout &lt;&lt; "Al doilea numar: ";     cin &gt;&gt; b;     int cif_unitati = a % 10;     int cif_zeci = (b / 10) % 10;     int suma=cif_unitati+cif_zeci;     cout &lt;&lt;"Suma cifre iunitatilor primului numar si a cifrei zecilor celui de-al doilea este: " &lt;&lt; suma &lt;&lt; endl;     return 0; } </pre> |

32. Scrieți un program în Python/C++, care citește de la tastatură un număr natural nenul n, cu exact patru cifre și afișează numărul după ce s-a interschimbat prima cu cifră cu ultima lui cifră.

**De exemplu**, pentru n=1234, se va afișa: 4231

### Limbajul Python

```
n = int(input("Introdu un numar de exact 4 cifre: "))
Extragem cifrele
prima = n // 1000
mij1 = (n // 100) % 10
mij2 = (n // 10) % 10
ultima = n % 10
Construim noul număr cu prima și ultima cifră inversate
noul_numar = ultima * 1000 + mij1 * 100 + mij2 * 10 + prima
print("Numărul după interschimbare este:", noul_numar)
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;

int main() {
 int n;
 cout << "Introdu un numar de exact 4 cifre: ";
 cin >> n;
 int prima = n / 1000;
 int mij1 = (n / 100) % 10;
 int mij2 = (n / 10) % 10;
 int ultima = n % 10;
 int noul_numar = ultima * 1000 + mij1*100 + mij2*10 + prima;
 cout<<"Nr.dupa interschimbare este: "<< noul_numar << endl;
 return 0;
}
```

33. Într-un ținut misterios, două ceasuri fermecate pornesc la drum din același loc, dar cu ritmuri diferite. Fiecare ceas îți transmite, la finalul cursei, cât timp a mers, exprimat în ore, clipe și bătăi. În acest ținut o oră are 60 de clipe iar o clipă are 70 de bătăi. Ceasurile îți spun valorile lor pe aceeași linie, în următoarea ordine: ore1, clipe1, batai1 pentru primul ceas și pentru cel de-al doilea ceas ore2, clipe2, batai2. Scrieți un program în **Python/C++**, care calculează timpul total cumulat.

**De exemplu**, pentru 2 45 60 și 1 20 50,

programul va afișa 4 6 40 ( bătăi:  $60 + 50 = 110 \rightarrow 1$  clipă și 40 bătăi.  
clipe:  $45 + 20 + 1 = 66 \rightarrow 1$  oră și 6 clipe  
ore:  $2 + 1 + 1 = 4$ )

### Limbajul Python

```
o1, c1, b1, o2, c2, b2 = map(int, input().split())
Adunăm bătăile
b_total = b1 + b2
c_total = c1 + c2 + b_total // 70
o_total = o1 + o2 + c_total // 60
Normalizăm
```

```

b_total %= 70
c_total %= 60
Afișăm timpul total
print(f"{o_total} {c_total} {b_total}")

```

#### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main() {
 int o1, c1, b1, o2, c2, b2, o_tot, b_tot, c_tot;
 cin >> o1 >> c1 >> b1 >> o2 >> c2 >> b2;
 b_tot = b1 + b2;
 c_tot = c1 + c2 + b_tot / 70; o_tot = o1 + o2 + c_tot / 60;
 b_tot %= 70; c_tot %= 60;
 cout << o_tot << " " << c_tot << " " << b_tot << endl;
 return 0;
}

```

34. Scrieți un program în Python/C++, care citește de la tastatură un număr natural  $n$ , format din exact două cifre diferite. Fiecare cifră a numărului este transformată astfel: dacă cifra este pară, se adună la cifra respectivă câtul împărțirii la 3, iar dacă este impară se adună dublul ultimei cifre a pătratului său, iar dacă rezultatul depășește 9, se ia doar ultima cifră a sa. Programul afișează pe ecran noul număr format din cele două cifre transformate, în aceeași ordine.

**De exemplu,** pentru  $n=37$ , programul va afișa **15** ( $a = 3$  (impar)  $\rightarrow 3 + 2 * (9 \% 10) = 21 \rightarrow 1$ ;  $b = 7$  (impar)  $\rightarrow 7 + 2 * (49 \% 10) = 25 \rightarrow 5$ ) iar pentru  $n=24$ , programul va afișa **45** ( $a = 2$  (par)  $\rightarrow 2 + 2 / 3 = 4$ ;  $b = 4$  (par)  $\rightarrow 4 + 1 = 5$ )

#### Limbajul Python

```

n = int(input())
a = n // 10 # cifra zecilor
b = n % 10 # cifra unitatilor
transformare pentru cifra a
a_t = (a + a // 3) * (a % 2 == 0) + (a + 2 * ((a * a) % 10)) * (a % 2 != 0)
transformare pentru cifra b
b_t = (b + b // 3) * (b % 2 == 0) + (b + 2 * ((b * b) % 10)) * (b % 2 != 0)
daca e mai mare de 9, luam doar ultima cifra
a_t %= 10
b_t %= 10
afisam noul numar
print(f"{a_t}{b_t}")

```

#### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{ int n, a_t, b_t, a, b;
 cin >> n;

```

```

a = n / 10; b = n % 10;
a_t=(a%2== 0)*(a + a / 3)+(a % 2 != 0)*(a + 2*((a * a)% 10));
b_t=(b % 2 == 0)*(b + b / 3)+(b % 2 != 0)*(b+2*((b * b)% 10));
a_t %= 10; b_t %= 10;
cout << a_t << b_t << endl;
return 0;
}

```

35. Scrieți un program care să citească de la tastatură un număr natural  $a$ , apoi să se afișeze, pe ecran, ultima cifră a numărului  $2^a$ .

De exemplu, pentru  $a=31$  se va afișa: 8.

| Limbajul Python                                                                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a = int(input("a= ")) if a==0:     print(1) else:     if a%4==0:         print(6)     if a%4==1:         print(2)     if a%4==2:         print(4)     if a%4==3:         print(8) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a;     cout&lt;&lt;"a="; cin&gt;&gt;a;     if(a==0)         cout&lt;&lt;1;     else     {         if(a%4==0) cout&lt;&lt;6;         if(a%4==1) cout&lt;&lt;2;         if(a%4==2) cout&lt;&lt;4;         if(a%4==3) cout&lt;&lt;8;     }     return 0; } </pre> |

### III. Structura alternativă

1. Scrieți un program **Python/C++** care citește de la tastatură, de pe linii separate, două numere întregi **a** și **b**, și apoi afișează cel mai mare număr dintre cele citite.

De exemplu, pentru **a=123** și **b=45**, programul afișează valoarea **123**.

| Limbajul Python                                                                          | Limbajul C/C++                                                                                                                                                                       |
|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a = int(input()) b = int(input()) if a&gt;b :     print(a) else:     print(b)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int a,b;     cin&gt;&gt;a&gt;&gt;b;     if(a&gt;b) cout&lt;&lt;a;     else cout&lt;&lt;b;     return 0; }</pre> |

2. Scrieți un program **Python/C++** citește de la tastatură un număr real **x** și apoi afișează mesajul **"POZITIV"** dacă valoarea lui **x** este pozitivă, **"NEGATIV"** dacă este negativă, sau **"NUL"** dacă este 0.

De exemplu, pentru **x=-17.23**, programul afișează mesajul **NEGATIV**.

| Limbajul Python                                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x = float(input()) if x&gt;0 :     print("POZITIV") else:     if x&lt;0:         print("NEGATIV")     else:         print("NUL")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     float x;     cin&gt;&gt;x;     if(x&gt;0) cout&lt;&lt;"POZITIV";     else if (x&lt;0) cout&lt;&lt;"NEGATIV";     else cout&lt;&lt;"NUL";     return 0; }</pre> |

3. Scrieți un program **Python/C++** care citește de la tastatură, de pe aceeași linie, trei numere reale distincte, apoi calculează și afișează pe ecran suma dintre cel mai mare și cel mai mic dintre cele trei citite. De exemplu, dacă se citesc valorile **3.16**, **-1**, **8.3**, programul afișează suma: **7.3** (**=-1+8.3**).

| Limbajul Python                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x,y,z = map(float, input().split()) maxi=x if maxi&lt;y : maxi=y if maxi&lt;z : maxi=z mini=x if mini&gt;y : mini=y if mini&gt;z : mini=z print(mini+maxi)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     float x,y,z,mini,maxi;     cin&gt;&gt;x&gt;&gt;y&gt;&gt;z;     maxi=x;     if(maxi&lt;y) maxi=y;     else if(maxi&lt;z)maxi=z;     mini=x;     if(mini&gt;y) mini=y;     else if(mini&gt;z)mini=z;     cout&lt;&lt;mini+maxi;     return 0; }</pre> |

4. Scrieți un program **Python/C++** care citește de la tastatură, de pe aceeași linie, separate prin câte un spațiu, două numere întregi și un caracter din mulțimea `{'+', '-', '*', '/'}`. În funcție de valoarea caracterului, programul va efectua operația aritmetică indicată între cele două numere întregi și va afișa pe ecran valoarea calculată.

De exemplu, dacă se citesc valorile `4 5 *`, programul va efectua operația de înmulțire a celor două numere și va afișa valoarea `20 (=4*5)`

| Limbajul Python                                                                                                                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x,y,op = input().split() x = int(x) y = int(y) if op == '+' :     print(x+y) elif op == '-' :     print(x-y) elif op == '*' :     print(x*y) elif op == '/' :     print(x/y) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int x,y;     char op;     cin&gt;&gt;x&gt;&gt;y&gt;&gt;op;     if (op=='+') cout&lt;&lt;x+y;     else if (op=='-') cout&lt;&lt;x-y;     else if (op=='*') cout&lt;&lt;x*y;     else if (op=='/')         cout&lt;&lt;(float)x/y;     return 0; } </pre> |

5. Scrieți un program **Python/C++** care citește de la tastatură un număr natural `n` format din exact 3 cifre. Programul afișează pe ecran mesajul **ORDONAT** dacă cifrele lui `n` sunt în ordine crescătoare, adică cifra sutelor este mai mică decât cifra zecilor, iar cifra zecilor este mai mică decât cifra unităților. Altfel, programul va afișa mesajul **NU**.

De exemplu, pentru `n=726`, programul va afișa **NU**, iar pentru `n=123`, programul va afișa **ORDONAT**.

| Limbajul Python                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) u=n%10 n=n//10 z=n%10 s=n//10 if s&lt;=z and z&lt;=u :     print("ORDONAT") else: print("NU") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,s,z,u;     cin&gt;&gt;n;     u=n%10; n=n/10;     z=n%10; s=n/10;     if (s&lt;=z &amp;&amp; z&lt;=u) cout&lt;&lt;"ORDONAT";     else cout&lt;&lt;"NU";     return 0; } </pre> |

6. Un elev are **S** lei pentru a cumpăra, de la librărie, trei caiete și trei coperti. El vrea să-i rămână cât mai mulți bani de buzunar după ce își cumpără aceste rechizite. La librărie se vând două tipuri de caiete (de **c1** lei și **c2** lei), trei tipuri de coperti (de **p1** lei, **p2** lei și **p3** lei) și două tipuri de seturi formate dintr-un caiet și o copertă (de **s1** lei și **s2** lei).

Scrieți un program **Python/C++** care să citească numerele naturale **S**, **c1**, **c2**, **p1**, **p2**, **p3**, **s1** și **s2**, și apoi să determine ce trebuie să cumpere elevul pentru a cheltui cât mai puțini bani. Programul va afișa o valoare reprezentând bani rămași.

De exemplu, dacă se citesc numerele: `110 27 17 19 23 16 41 37`, elevul va alege 3 caiete a câte 17 lei și 3 coperti a câte 16 lei. Programul va afișa `11`, reprezentând banii rămași ( $110 - 3 \cdot 17 - 3 \cdot 16 = 11$ ).

| Limbajul Python                                                                                                                                                                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> S,c1,c2,p1,p2,p3,s1,s2 =     map(int,input().split()) if c1&lt;c2 : minc=c1 else: minc=c2 if p1&lt;p2 : minp=p1 else: minp=p2 if minp&gt;p3 : minp=p3 if s1&lt;s2 : mins=s1 else: mins=s2 if minc+minp&lt;mins :     rest=S-3*(minc+minp) else: rest=S-3*mins; print(rest) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int S,c1,c2,p1,p2,p3;     int s1, s2, rest;     int minc, minp, mins;     cin&gt;&gt;S&gt;&gt;c1&gt;&gt;c2;     cin&gt;&gt;p1&gt;&gt;p2&gt;&gt;p3&gt;&gt;s1&gt;&gt;s2;     if(c1&lt;c2)minc=c1;     else minc=c2;     if(p1&lt;p2)minp=p1;     else minp=p2;     if(minp&gt;p3) minp=p3;     if(s1&lt;s2)mins=s1;     else mins=s2;     if(minc+minp&lt;mins)         rest=S-3*(minc+minp);     else rest=S-3*mins;     cout&lt;&lt;rest;     return 0; } </pre> |

7. Scrieți un program Python/C++ care citește un număr natural nenul  $S$  ( $S \leq 100000$ ) și care determină și afișează patru numere naturale  $x, y, z$  și  $w$  astfel încât  $x+y+z+w=S$  și produsul  $x*y*z*w$  este maximul dintre produsele oricăror patru numere naturale a căror suma este  $S$ .

De exemplu, pentru  $n=18$ , programul va afișa 4 4 5 5.

| Limbajul Python                                                                                                                                                                                                        | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> S=int(input()) x=y=z=w=S//4 r=S%4  if r==1 :     w=w+1 else :     if r==2 :         w=w+1         z=z+1     else :         if r==3 :             w=w+1             z=z+1             y=y+1 print(x,y,z,w) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int S,x,y,z,w,r,a;     cin&gt;&gt;S;     x=y=z=w=S/4; r=S%4;     if(r==1)w++;     else if(r==2)         {w++; z++;}     else if(r==3)         {w++; z++; y++;}      cout&lt;&lt;x&lt;&lt;" "&lt;&lt;y&lt;&lt;" ";     cout&lt;&lt;z&lt;&lt;" "&lt;&lt;w;     return 0; } </pre> |

8. Scrieți un program Python/C++ care citește de la tastatură un număr natural  $n$  și verifică dacă este un pătrat perfect. Programul afișează pe ecran mesajul DA dacă  $n$  este pătrat perfect, respectiv NU în caz contrar. De exemplu, pentru  $n=141$ , programul va afișa NU, iar pentru  $n=121$ , programul va afișa DA.

| Limbajul Python                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import math n = int(input()) radn = int(math.sqrt(n)) if radn*radn==n : print("DA") else : print("NU")</pre> | <pre>#include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std; int main() {     int n,radn;     cin&gt;&gt;n;     radn=sqrt(n);     if(n==radn*radn)         cout&lt;&lt;"DA";     else cout&lt;&lt;"NU";     return 0; }</pre> |

9. Un melc trebuie să parcurgă o distanță de  $D$  metri. El se deplasează cu o viteză de  $v$  km/h. În câte minute va parcurge melcul distanța  $D$ ?

Scrieți un program **Python/C++** care să citească două numere reale  $D$  și  $v$ , de pe aceeași linie, și apoi să determine numărul de minute necesare melcului pentru a parcurge distanța  $D$ .

De exemplu, pentru  $v=0.3$  km/h,  $d=20$  programul afișează 4 (=4 minute)

| Limbajul Python                                                                                 | Limbajul C/C++                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>V,D=map(float, input().split()) D=D/1000 M=int(60*D/V) if M!=60*D/V : M=M+1 print(M)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     float V,D;     int M;     cin&gt;&gt;V&gt;&gt;D;     D=D/1000; M=60*D/V;     if(M!=60*D/V) M++;     cout&lt;&lt;M;     return 0; }</pre> |

10. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  și verifică dacă  $n$  se poate scrie ca o sumă de trei numere naturale consecutive. În acest caz, programul afișează pe ecran cele trei numere naturale consecutive, separate prin câte un spațiu. În caz contrar va afișa mesajul **NU**.

De exemplu, pentru  $n=24$  se afișează: 7 8 9, iar pentru  $n=11$  se afișează **NU**.

| Limbajul Python                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) if n%3==0 :     x=n//3     print(x-1, x, x+1) else: print("NU")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n,x; //n=x-1+x+x+1=3*x     cin&gt;&gt;n;     if(n%3==0) { x=n/3;         cout&lt;&lt;x-1&lt;&lt;" "&lt;&lt;x&lt;&lt;" "&lt;&lt;x+1; }     else cout&lt;&lt;"NU";     return 0; }</pre> |

11. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $9 < n < 10^9$ ) și care afișează cel mai mare număr care se formează cu cifra unităților și cifra zecilor din  $n$ . Programul nu va utiliza funcții predefinite (de exemplu max).

Exemplu: pentru  $n=574017$  se afișează 71

| Limbajul Python                                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) u = n % 10 z = (n // 10) % 10 n1 = z * 10 + u n2 = u * 10 + z if n1 &gt; n2:     print(n1) else:     print(n2) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned n;     cin&gt;&gt;n;     int u=n%10;     int z=(n/10)%10;     int n1=z*10+u;     int n2=u*10+z;     if (n1&gt;n2)         cout&lt;&lt;n1;     else         cout&lt;&lt;n2;     return 0; } </pre> |

12. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $99 < n < 1000$ ) și care afișează cifrele numărului  $n$  în ordine crescătoare, separate printr-un spațiu. Programul nu va utiliza funcții predefinite (de exemplu max).

Exemplu: pentru  $n=207$  se afișează 0 2 7

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                        | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) a = n // 100 b = (n // 10) % 10 c = n % 10 if a &lt;= b and a &lt;= c:     if b &lt;= c:         print(a, b, c)     else:         print(a, c, b) elif b &lt;= a and b &lt;= c:     if a &lt;= c:         print(b, a, c)     else:         print(b, c, a) else:     if a &lt;= b:         print(c, a, b)     else:         print(c, b, a) </pre> | <pre> #include &lt;iostream&gt; using namespace std;  int main() {     int n;     cin&gt;&gt;n;     int a=n/100;     int b=(n/10)%10;     int c=n%10;     int aux;     if(a&gt;b)     {         aux=a; a=b; b=aux;     }     if(a&gt;c)     {         aux=a; a=c; c=aux;     }     if(b&gt;c)     {         aux=b; b=c; c=aux;     }     cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;" "&lt;&lt;c;     return 0; } </pre> |

13. Scrieți un program **Python/C++** care citește un număr natural nenul,  $n$ . Dacă acesta este par se vor citi de pe același rând două numere reale și se va afișa maximul dintre ele iar dacă este impar se vor citi trei numere întregi de pe rânduri diferite și se va afișa cifra maximă a lor. Toate numerele citite aparțin intervalului  $[100, 999]$ . Programul nu va utiliza funcții predefinite (de exemplu max).

Exemplu: pentru  $n=103$  se citesc 345 612 803 și se afișează 8 iar pentru  $n=586$  se citesc 314.5 640.12 și se afișează 640.12

#### Limbajul Python

```
n = int(input())
if n % 2 == 0:
 # citire numere de pe același rând
 x, y = map(float, input().split())
 if x > y:
 print(x)
 else:
 print(y)
else:
 # citire numere de pe rânduri diferite
 a = int(input())
 b = int(input())
 c = int(input())
 a1, a2, a3 = a // 100, (a // 10) % 10, a % 10
 b1, b2, b3 = b // 100, (b // 10) % 10, b % 10
 c1, c2, c3 = c // 100, (c // 10) % 10, c % 10
 max_cifra = a1
 if a2 > max_cifra: max_cifra = a2
 if a3 > max_cifra: max_cifra = a3
 if b1 > max_cifra: max_cifra = b1
 if b2 > max_cifra: max_cifra = b2
 if b3 > max_cifra: max_cifra = b3
 if c1 > max_cifra:
 max_cifra = c1
 if c2 > max_cifra:
 max_cifra = c2
 if c3 > max_cifra:
 max_cifra = c3
 print(max_cifra)
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n;
 cin >> n;
 if (n % 2 == 0)
 {
 double x, y;
 cin >> x >> y;
 cout << (x > y ? x : y);
 }
 else
 {

```

```

 int a,b,c;
 cin>>a>>b>>c;
 int a1=a/100;
 int a2=a/10%10;
 int a3=a%10;
 int b1=b/100;
 int b2=b/10%10;
 int b3=b%10;
 int c1=c/100;
 int c2=c/10%10;
 int c3=c%10;
 int maxCif=a1;
 if(a2>maxCif) maxCif=a2;
 if(a3>maxCif) maxCif=a3;
 if(b1>maxCif) maxCif=b1;
 if(b2>maxCif) maxCif=b2;
 if(b3>maxCif) maxCif=b3;
 if(c1>maxCif) maxCif=c1;
 if(c2>maxCif) maxCif=c2;
 if(c3>maxCif) maxCif=c3;
 cout<<maxCif;
}
return 0;
}

```

14. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n_1$  ( $9 < n_1 < 100$ ) și, de pe același rând un număr natural  $n_2$  ( $99 < n_2 < 1000$ ). Programul afișează cel mai mic număr cu 2 cifre care se obține folosind o cifră din primul număr și o cifră din al doilea număr.

Exemplu: pentru  $n_1=62$  și  $n_2=801$  atunci se afișează 12.

#### Limbajul Python

```

n1, n2 = map(int, input().split())
c1 = n1 // 10
c2 = n1 % 10
d1 = n2 // 100
d2 = (n2 // 10) % 10
d3 = n2 % 10
minim = 99
def verifica(num):
 global minim
 if num >= 10 and num < minim:
 minim = num
verifica(c1 * 10 + d1)
verifica(d1 * 10 + c1)
verifica(c1 * 10 + d2)
verifica(d2 * 10 + c1)
verifica(c1 * 10 + d3)
verifica(d3 * 10 + c1)
verifica(c2 * 10 + d1)
verifica(d1 * 10 + c2)
verifica(c2 * 10 + d2)
verifica(d2 * 10 + c2)
verifica(c2 * 10 + d3)
verifica(d3 * 10 + c2)
print(minim)

```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
bool esteValid(int num)
{ return num >= 10; }

int main()
{
 int n1,n2;
 cin>>n1>>n2;
 int c1=n1/10;
 int c2=n1%10;
 int d1=n2/100;
 int d2=n2/10%10;
 int d3=n2%10;
 int minim=99;
 int num;
 num=c1*10+d1;
 if(esteValid(num)&&num<minim) minim=num;
 num=d1*10+c1;
 if(esteValid(num)&&num<minim) minim=num;
 num=c1*10+d2;
 if(esteValid(num)&&num<minim) minim=num;
 num=d2*10+c1;
 if(esteValid(num)&&num<minim) minim=num;
 num=c1*10+d3;
 if(esteValid(num)&&num<minim) minim=num;
 num=d3*10+c1;
 if(esteValid(num)&&num<minim) minim =num;
 num=c2*10+d1;
 if (esteValid(num)&&num<minim) minim=num;
 num=d1*10+c2;
 if(esteValid(num)&&num<minim) minim=num;
 num=c2*10+d2;
 if(esteValid(num)&&num<minim) minim=num;
 num=d2*10+c2;
 if(esteValid(num)&&num<minim) minim=num;
 num=c2*10+d3;
 if(esteValid(num)&&num<minim) minim=num;
 num=d3*10+c2;
 if (esteValid(num)&&num<minim) minim=num;
 cout<<minim;
 return 0;
}
```

15. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, trei numere reale  $a, b$  și  $c$  ( $0 < a, b, c < 10^3$ ). Programul afișează mesajul **triunghi** dacă cele trei numere scrise alăturat formează laturile unui triunghi, situație în care se afișează tipul de triunghi (mesaj de forma: **echilateral**, **isoscel**, **oarecare**) sau mesajul **nu** în caz contrar.

Exemplu: pentru  $a=5$ ,  $b=6$  și  $c=5$  atunci se afișează: **isoscel**

### Limbajul Python

```
a, b, c = map(float, input().split())
if a + b > c and a + c > b and b + c > a:
 print("triunghi")
 if a == b == c:
 print("echilateral")
 elif a == b or b == c or a == c:
 print("isoscel")
 else:
 print("oarecare")
else:
 print("nu")
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 double a,b,c;
 cin>>a>>b>>c;
 if (a+b>c&& a+c>b&& b+c>a)
 {
 cout<<"triunghi"<<'\\n';
 if (a==b&& b==c)
 cout<<"echilateral";
 else if (a==b || b==c || a==c)
 cout<<"isoscel";
 else cout<<"oarecare";
 }
 else cout<<"nu";
 return 0;
}
```

16. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, trei numere naturale nenule **zi**, **luna** și **an** reprezentând o dată calendaristică. Programul afișează numărul de zile rămase până la sfârșitul lunii.

Exemplu: dacă se citește 17 8 2025 atunci se afișează 14.

### Limbajul Python

```
def este_bisect(an):
 return (an % 4 == 0 and an % 100 != 0) or (an % 400 == 0)
zi, luna, an = map(int, input().split())
if luna == 2:
 zile_luna = 29 if este_bisect(an) else 28
elif luna == 4 or luna == 6 or luna == 9 or luna == 11:
 zile_luna = 30
else: zile_luna = 31
zile_ramase = zile_luna - zi
print(zile_ramase)
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
bool esteBisect(int an)
{
 return (an%4==0&&an%100!=0) || (an%400==0);
}
int main()
{
 int zi, luna, an;
 cin>>zi>>luna>>an;
 int zileLuna;
 if(luna==2)
 zileLuna=esteBisect(an)?29:28;
 else if(luna==4||luna==6||luna==9||luna==11)
 zileLuna=30;
 else
 zileLuna=31;
 int zileRamase=zileLuna-zi;
 cout<<zileRamase;
 return 0;
}
```

17. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, trei numere naturale nenule **zi**, **luna** și **an** reprezentând o dată calendaristică. Programul afișează numărul de zile de la începutul anului până la ziua curentă inclusiv. Exemplu: dacă se citește 19 3 2024 atunci se afișează 79.

### Limbajul Python

```
def este_bisect(an):
 return (an % 4 == 0 and an % 100 != 0) or (an % 400 == 0)
zi, luna, an = map(int, input().split())
zile_totale = zi
if luna > 1: zile_totale += 31
if luna > 2:
 if este_bisect(an): zile_totale += 29
 else: zile_totale += 28
if luna > 3: zile_totale += 31
if luna > 4: zile_totale += 30
if luna > 5: zile_totale += 31
if luna > 6: zile_totale += 30
if luna > 7: zile_totale += 31
if luna > 8: zile_totale += 31
if luna > 9: zile_totale += 30
if luna > 10: zile_totale += 31
if luna > 11: zile_totale += 30
print(zile_totale)
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
bool esteBisect(int an)
{ return (an%4==0 && an%100!=0) || (an%400==0); }
```

```

int main()
{
 int zi, luna, an;
 cin>>zi>>luna>>an;
 int zileTotale=zi;
 if(luna>1) zileTotale+=31;
 if(luna>2) zileTotale+=esteBisect(an)?29:28;
 if(luna>3) zileTotale+=31;
 if(luna>4) zileTotale+=30;
 if(luna>5) zileTotale+=31;
 if(luna>6) zileTotale+=30;
 if(luna>7) zileTotale+=31;
 if(luna>8) zileTotale+=31;
 if(luna>9) zileTotale+=30;
 if(luna>10) zileTotale+=31;
 if(luna>11) zileTotale+=30;
 cout<<zileTotale;
 return 0;
}

```

18. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, trei numere naturale nenule **zi**, **luna** și **an** reprezentând o dată calendaristică. Programul afișează numărul de zile rămase până la sfârșitul anului.

Exemplu: dacă se citește 17 11 2024 atunci se afișează 44.

#### Limbajul Python

```

def este_bisect(an):
 return (an % 4 == 0 and an % 100 != 0) or (an % 400 == 0)
zi, luna, an = map(int, input().split())
zile_in_an = 366 if este_bisect(an) else 365
zile_trecute = zi
if luna > 1: zile_trecute += 31
if luna > 2: zile_trecute += 29 if este_bisect(an) else 28
if luna > 3: zile_trecute += 31
if luna > 4: zile_trecute += 30
if luna > 5: zile_trecute += 31
if luna > 6: zile_trecute += 30
if luna > 7: zile_trecute += 31
if luna > 8: zile_trecute += 31
if luna > 9: zile_trecute += 30
if luna > 10: zile_trecute += 31
if luna > 11: zile_trecute += 30
zile_ramase = zile_in_an - zile_trecute
print(zile_ramase)

```

#### Limbajul C/C++

```

#include <iostream>
using namespace std;
bool esteBisect(int an)
{ return (an%4==0 && an%100!=0) || (an%400==0); }
int main()
{
 int zi, luna, an;
 cin>>zi>>luna>>an;

```

```

 int zileInAn=esteBisect(an)?366:365;
 int zileTrecute=zi;
 if(luna>1) zileTrecute+=31;
 if(luna>2) zileTrecute+=esteBisect(an)?29:28;
 if(luna>3) zileTrecute+=31;
 if(luna>4) zileTrecute+=30;
 if(luna>5) zileTrecute+=31;
 if(luna>6) zileTrecute+=30;
 if(luna>7) zileTrecute+=31;
 if(luna>8) zileTrecute+=31;
 if(luna>9) zileTrecute+=30;
 if(luna>10) zileTrecute+=31;
 if(luna>11) zileTrecute+=30;
 int zileRamase=zileInAn-zileTrecute;
 cout<<zileRamase;
 return 0;
}

```

19. Scrieți un program **Python/C++** care citește un număr real  $x$  și afișează rezultatul expresiei alăturat.

Obs. Trebuie avut în vedere condițiile de existență al radicalului!

Numărul de zecimale este cel implicit al limbajului.

Exemplu:

dacă  $x=10$  se afișează:  $E(10)=-0.375000$

$$E(x)=\begin{cases} x^2+3\cdot x-2, & \text{dacă } x\leq 2 \\ \sqrt{x}-\frac{x}{2}, & \text{dacă } 2<x<5 \\ \frac{x+1}{2-x}+1, & \text{dacă } 5\leq x<100 \\ 2\cdot x+\sqrt{x-2^x}, & \text{altfel} \end{cases}$$

### Limbajul Python

```

import math
x = float(input())
if x <= 2: e = x**2 + 3*x - 2
elif 2 < x < 5: e = math.sqrt(x) - x / 2
elif 5 <= x < 100: e = (x + 1) / (2 - x) + 1
else:
 putere = 2 ** x
 dif = x - putere
 if dif < 0:
 print("Eroare")
 exit()
 e = 2 * x + math.sqrt(dif)
print("E(",x,")=",f"{e:.6f}")

```

### Limbajul C/C++

```

#include <iostream>
#include <cmath>
using namespace std;
int main()
{
 double x,E;
 cin>>x;
 if(x<=2) E=x*x+3*x-2;
 else if(x>2&&x<5) E=sqrt(x)-x/2;
 else if(x>=5&&x<100) E=(x+1)/(2-x)+1;
 else

```

```

 { double putere=pow(2,x);
 double dif=x-putere;
 if(dif<0) { cout<<"Eroare"; return 0; }
 E=2*x+sqrt(dif);
 }
 cout<<"E("<<x<<")="<<E;
 return 0;
}

```

20. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, trei numere reale  $a$ ,  $b$  și  $c$  ( $a, b, c < 10^4$ ). Programul rezolvă ecuația de gradul al II-lea,  $a \cdot x^2 + b \cdot x + c = 0$ , și afișează, pe rânduri diferite, soluțiile ecuației în ordine crescătoare, ca în exemplu. Numărul de zecimale este cel implicit al limbajului.

**Exemplu:** pentru  $a=1$ ,  $b=-5$  și  $c=6$  atunci se afișează

$x_1=2$

$x_2=3$

### Limbajul Python

```

import math
a, b, c = map(float, input().split())
if a == 0:
 if b == 0:
 print("Ecuatia nu are solutii reale.")
 else:
 x = -c / b
 print(f"x1={x}")
else:
 delta = b**2 - 4*a*c
 if delta < 0:
 print("Ecuatia nu are solutii reale.")
 elif delta == 0:
 x = -b / (2*a)
 print(f"x={x}")
 else:
 sqrt_delta = math.sqrt(delta)
 x1 = (-b - sqrt_delta) / (2*a)
 x2 = (-b + sqrt_delta) / (2*a)
 if x1 < x2:
 print(f"x1={x1}")
 print(f"x2={x2}")
 else:
 print(f"x1={x2}")
 print(f"x2={x1}")

```

### Limbajul C/C++

```

#include <iostream>
#include <cmath>
using namespace std;
int main()
{ double a,b,c;
 cin>>a>>b>>c;
 if(a==0)
 {
 if(b==0)
 cout<<"Ecuatia nu are solutii reale.";
 }
}

```

```

 else
 {
 double x=-c/b;
 cout<<"x1="<<x;
 }
 }
 else
 {
 double delta=b*b-4*a*c;
 if(delta<0) cout<<"Ecuatia nu are solutii reale.";
 else if(delta==0)
 {
 double x=-b/(2*a); cout<<"x="<<x;
 }
 else
 {
 double sqrt_delta=sqrt(delta);
 double x1=(-b-sqrt_delta)/(2*a);
 double x2=(-b+sqrt_delta)/(2*a);
 if(x1>x2)
 swap(x1, x2);
 cout<<"x1="<<x1<< '\n';
 cout<<"x2="<<x2;
 }
 }
 return 0;
}

```

21. Scrieți un program **Python/C++**, care citește de la tastatură, de pe linii separate, două numere întregi **a** și **b**, cu cel mult patru cifre fiecare,  $a < b$ , apoi afișează, pe ecran, natura produsului numerelor întregi din intervalul  $[a, b]$ , astfel: mesajul **STRICT POZITIV**, dacă produsul numerelor este strict pozitiv, mesajul **STRICT NEGATIV**, dacă produsul numerelor este strict negativ sau mesajul **NUL**, dacă produsul numerelor este nul.

**De exemplu**, dacă  $a=-4$  și  $b=8$ , programul afișează mesajul **NUL**

- a) dacă  $a=-4$  și  $b=-1$ , programul afișează mesajul **STRICT POZITIV**  
 b) dacă  $a=-4$  și  $b=-2$ , programul afișează mesajul **STRICT NEGATIV**

| Limbajul Python                                                                                                                                                                                          | Limbajul C/C++                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a = int(input()) b = int(input()) if a&lt;=0 and b&gt;=0 :     print("NUL") else:     if b&lt;0 and (b-a+1)%2!=0:         print("STRICT NEGATIV")     else:         print("STRICT POZITIV") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a,b;     cin&gt;&gt;a&gt;&gt;b;     if(a&lt;=0 and b&gt;=0)         cout&lt;&lt;"NUL";     else         if (b&lt;0 and (b-a+1)%2!=0)             cout&lt;&lt;"STRICT NEGATIV";         else cout&lt;&lt;"STRICT POZITIV";     return 0; } </pre> |

22. Scrieți un program **Python/C++**, care citește de la tastatură, de pe linii separate, patru numere reale **a**, **b**, **c** și **d**, cu cel mult patru cifre fiecare,  $a < b$  și  $c < d$ , apoi afișează,

pe ecran, mesajul **DA**, dacă intersecția intervalului deschis la ambele capete  $(a, b)$  cu intervalul deschis la ambele capete  $(c, d)$  este mulțimea vidă sau mesajul **NU**, în caz contrar.

**De exemplu**, dacă  $a=-2.5$   $b=4.3$   $c=5$  și  $d=8$ , programul afișează mesajul **DA** iar dacă  $a=-2.5$   $b=4.3$   $c=4$  și  $d=8$ , programul afișează mesajul **NU**

| Limbajul Python                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a = float(input()) b = float(input()) c = float(input()) d = float(input()) if b&lt;=c or d&lt;=a:     print("DA") else:     print("NU") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     float a,b,c,d;     cin&gt;&gt;a&gt;&gt;b&gt;&gt;c&gt;&gt;d;     if(b&lt;=c or d&lt;=a)         cout&lt;&lt;"DA";     else cout&lt;&lt;"NU";     return 0; } </pre> |

23. Scrieți un program **Python/C++**, care citește de la tastatură, de pe aceeași linie, trei numere întregi  $m$ ,  $n$  și  $p$ , cu cel mult patru cifre fiecare, apoi afișează, pe ecran, cifra de control a modulului sumei celor trei numere. Cifra de control a unui număr se obține efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, până când suma obținută este un număr format dintr-o singură cifră. Această ultimă cifră se numește cifră de control. De exemplu, dacă se citesc valorile 20 -100 1, programul afișează: 7 deoarece  $20+(-100)+1$  este -79, modulul numărului -79 este 79, iar cifra de control a lui 79 este 7.

| Limbajul Python                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> m,n,p = map(int, input().split()) s=m+n+p if s&lt;0: s=-s if s==0 or s%9!=0: cifc=s%9 else: cifc=9 print(cifc) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int m,n,p,s,cifc;     cin&gt;&gt;m&gt;&gt;n&gt;&gt;p;     s=m+n+p;     if (s&lt;0) s=-s;     if(s==0 or s%9!=0) cifc=s%9;     else cifc=9;     cout&lt;&lt;cifc;     return 0; } </pre> |

24. Scrieți un program **Python/C++**, care citește de la tastatură, de pe aceeași linie, separate prin câte un spațiu, trei numere naturale nenule  $a$ ,  $b$ ,  $c$ , unde  $a < b$ ,  $b < c$ , având cel mult patru cifre fiecare și un caracter  $op$  din mulțimea  $\{'a', 'g'\}$ . În funcție de valoarea caracterului  $op$ , programul afișează, pe ecran, mesajul **DA** dacă cele trei numere pot forma termenii consecutivi ai unei progresii aritmetice (pentru opțiunea 'a'), respectiv termenii consecutivi ai unei progresii geometrice (pentru opțiunea 'g') sau mesajul **NU**, în caz contrar.

**De exemplu**, dacă se citesc valorile 4 7 10 a, programul va afișa mesajul **DA** dacă se citesc valorile 4 7 10 g, programul va afișa mesajul **NU**, dacă se citesc valorile 2 4 8 g, programul va afișa mesajul **DA**

| Limbajul Python                                                                                                                                                        | Limbajul C/C++                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a, b, c, op = input().split() a = int(a) b = int(b) c = int(c) if (op == 'a' and 2*b==a+c) or (op == 'g' and b*b==a*c):     print("DA") else: print("NU") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a,b,c;     char op;     cin&gt;&gt;a&gt;&gt;b&gt;&gt;c&gt;&gt;op;     if ((op == 'a' and 2*b==a+c) or         (op == 'g' and b*b==a*c))         cout&lt;&lt;"DA";     else cout&lt;&lt;"NU";     return 0; } </pre> |

25. Scrieți un program **Python/C++**, care citește de la tastatură un număr natural  $n$ , având exact două cifre. Cifrele numărului  $n$  se modifică după următoarele reguli: dacă cifra numărului este cifră primă, cifra scade cu 1, dacă cifra numărului este cifră pătrat perfect, atunci cifra se înlocuiește cu rădăcina pătrată a cifrei, restul cifrelor nu se modifică. Programul afișează pe ecran numărul  $n$ , după transformarea cifrelor numărului. Dacă, după aplicarea regulilor enunțate, numărul  $n$  este egal cu numărul citit inițial, programul afișează mesajul **nu exista**.

De exemplu, pentru  $n=29$ , programul va afișa 13, iar pentru  $n=68$  sau pentru  $n=80$ , programul va afișa nu exista.

| Limbajul Python                                                                                                                                                                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> import math n = int(input()) a=n//10 b=n%10 rada = int(math.sqrt(a)) radb = int(math.sqrt(b)) if rada*rada==a :     a=rada if radb*radb==b :     b=radb if a==2 or a==3 or a==5 or a==7:     a=a-1 if a*10+b==n:     print("nu exista") else:     print(a*10+b) </pre> | <pre> #include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std; int main() {     int n,a,b,rada,radb;     cin&gt;&gt;n;     a=n/10; b=n%10;     rada=int(sqrt(a));     radb=int(sqrt(b));     if(rada*rada==a) a=rada;     if(radb*radb==b) b=radb;     if (a==2 or a==3 or a==5         or a==7) a=a-1;     if (a*10+b==n)         cout&lt;&lt;"nu exista";     else cout&lt;&lt;a*10+b;     return 0; } </pre> |

26. Scrieți un program **Python/C++** care citește de la tastatură o valoare întreagă care reprezintă ora (0–23) și afișează pe ecran mesajul:

- „dimineață” pentru orele 5–11,
- „după-amiază” pentru orele 12–17,
- „seară” pentru orele 18–22,
- „noapte”, în rest.

De exemplu, pentru ora = 6, programul va afișa: dimineață.

| 27. | Limbajul Python                                                                                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <pre> ora = int(input())  if 5 &lt;= ora &lt;= 11:     print("dimineață") elif 12 &lt;= ora &lt;= 17:     print("după-amiază") elif 18 &lt;= ora &lt;= 22:     print("seară") else:     print("noapte") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int ora;     cin &gt;&gt; ora;     if (ora &gt;= 5 &amp;&amp; ora &lt;= 11)         cout &lt;&lt; "dimineață";     else if (ora &gt;= 12 &amp;&amp; ora &lt;= 17)         cout &lt;&lt; "după-amiază";     else if (ora &gt;= 18 &amp;&amp; ora &lt;= 22)         cout &lt;&lt; "seară";     else cout &lt;&lt; "noapte";     return 0; } </pre> |

28. Scrieți un program **Python/C++** care citește de la tastatură coordonatele a două puncte diferite din plan, **A**( $x_1$ ,  $y_1$ ) și **B**( $x_2$ ,  $y_2$ ), și determină natura dreptei formate de acestea. Programul va afișa unul dintre următoarele mesaje:

- „dreaptă orizontală”, dacă dreapta este paralelă cu axa Ox,
- „dreaptă verticală”, dacă dreapta este paralelă cu axa Oy,
- „dreaptă oblică”, în orice alt caz.

Exemplu:

- Dacă A(1, 5) și B(4, 5) → dreapta orizontală
- Dacă A(2, 3) și B(2, 7) → dreapta verticală
- Dacă A(1, 1) și B(3, 4) → dreapta oblică

| 29. | Limbajul Python                                                                                                                                                                                                                            | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                      |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <pre> x1, y1 = map(int, input().split()) x2, y2 = map(int, input().split())  # Verificăm tipul dreptei if y1 == y2:     print("dreaptă orizontală") elif x1 == x2:     print("dreaptă verticală") else:     print("dreaptă oblică") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int x1, y1, x2, y2;     cin &gt;&gt; x1 &gt;&gt; y1;     cin &gt;&gt; x2 &gt;&gt; y2;     if (y1 == y2)         cout &lt;&lt; "dreaptă orizontală";     else if (x1 == x2)         cout &lt;&lt; "dreaptă verticală";     else         cout &lt;&lt; "dreaptă oblică";     return 0; } </pre> |

30. Să se scrie un program în **Python/C++** care citește de la tastatură un număr natural **c**, format din exact 13 cifre, reprezentând un cod de identificare spațială primit de un explorator intergalactic. Codul conține, în ordine, următoarele informații:

- prima cifră → sectorul spațial (1–9),
- următoarele două cifre → anul nașterii (ultimele două cifre),
- următoarele două cifre → luna nașterii,
- următoarele două cifre → ziua nașterii,
- restul cifrelor nu sunt relevante pentru această problemă.

Să se extragă din codul **c** anul, luna și ziua nașterii ale exploratorului și le afișează în formatul: **ZZ.LL.AAAA**

Pentru conversia completă a anului:

- dacă sectorul este 1 sau 2 → anul este în secolul 1900,
- dacă sectorul este 5 sau 6 → anul este în secolul 2000.

Exemplu: Pentru  $c = 1960501223457 \rightarrow \text{sector} = 1, \text{an} = 1960, \text{luna} = 05, \text{ziua} = 01$ . Data nașterii: 01.05.1960

### Limbajul Python

```
c = int(input())
Extragem componentele prin operații matematice
sector = c // 10**12
an = (c // 10**10) % 100
luna = (c // 10**8) % 100
zi = (c // 10**6) % 100
Determinăm anul complet
if sector == 1 or sector == 2: an_final = 1900 + an
elif sector == 5 or sector == 6: an_final = 2000 + an
else: an_final = -1
if an_final != -1:
 print(f"Data nașterii: {zi:02}.{luna:02}.{an_final}")
else: print("Cod invalid - sector necunoscut.")
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 long long c;
 cin >> c;
 int sector = c / 1000000000000, an = (c / 10000000000) % 100;
 int luna = (c / 100000000) % 100, zi = (c / 1000000) % 100;
 int an_final;
 if (sector == 1 || sector == 2) an_final = 1900 + an;
 else if (sector == 5 || sector == 6) an_final = 2000 + an;
 else { cout << "Cod invalid - sector necunoscut.\n";
 return 0; }
 cout << "Data nașterii: " << zi << ".";
 cout << luna << "." << an_final << endl;
 return 0; }
```

31. Scrieți un program în **Python/C++** care citește de la tastatură un număr natural  $N$  (cu cel mult 3 cifre), reprezentând un indiciu numeric primit de Făt-Frumos de la înțeleptul păzitor al Arsenalului Fermecat.

Făt-Frumos se pregătește să pornească într-o nouă aventură prin Pădurea de Aramă. Înainte de plecare, trebuie să aleagă sabia potrivită. Pentru a face alegerea, el aplică următorul algoritm magic, în funcție de  $N$ :

- dacă  $N$  are o singură cifră, sabia aleasă are numărul egal cu dublul lui  $N$ ,
- dacă  $N$  are două cifre, sabia aleasă are numărul egal cu produsul cifrelor lui  $N$ ,
- dacă  $N$  are trei cifre, sabia aleasă are numărul format din prima și ultima cifră ale lui  $N$ , în această ordine.

Programul va afișa numărul sabiei pe care trebuie să o aleagă Făt-Frumos.

**Exemple:**

- $N = 7 \rightarrow$  o cifră  $\rightarrow$  sabia:  $2 \times 7 = 14$
- $N = 46 \rightarrow$  două cifre  $\rightarrow$  sabia:  $4 \times 6 = 24$
- $N = 381 \rightarrow$  trei cifre  $\rightarrow$  sabia: 3 și 1  $\rightarrow$  31

### Limbajul Python

```
n = int(input())
if n < 10:
 rezultat = 2 * n
elif n < 100:
 cif1 = n // 10
 cif2 = n % 10
 rezultat = cif1 * cif2
elif n < 1000:
 prima = n // 100
 ultima = n % 10
 rezultat = prima * 10 + ultima
else:
 rezultat = -1
if rezultat != -1: print("Sabia aleasă are numărul:", rezultat)
else: print("Numărul N nu este valid.")
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n;
 cin >> n;
 int rezultat;
 if (n < 10) rezultat = 2 * n;
 else if (n < 100) {
 int cif1 = n / 10;
 int cif2 = n % 10;
 rezultat = cif1 * cif2; }
 else if (n < 1000) {
 int prima = n / 100; int ultima = n % 10;
 rezultat = prima * 10 + ultima; }
 else {
 cout << "Numărul N nu este valid." << endl;
 return 0; }
 cout << "Sabia aleasă are numărul: " << rezultat << endl;
 return 0;
}
```

32. Scrieți un program **Python/C++**, care citește de la tastatură, un număr natural  $n$ . Să se verifice dacă numărul are exact 3 cifre și dacă suma dintre cea mai mare și cea mai mică cifră a sa este egală cu cifra rămasă. Dacă numărul este special, adică suma dintre cea mai mare și cea mai mică cifră a sa este egală cu cifra rămasă atunci se afișează mesajul **special** altfel se va afișa mesajul **NU**

**De exemplu**, pentru  $n = 231$  programul afișează mesajul **special**  
 pentru  $n = 543$  programul afișează mesajul **NU**

| Limbajul Python                                                                                                                                                                                                                                                                                                          | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("n= ")) if 100 &lt;= n &lt;= 999:     c1 = n // 100     c2 = (n // 10) % 10     c3 = n % 10     cifre = [c1, c2, c3]     maxi = max(cifre)     mini = min(cifre)     suma = maxi + mini     med = sum(cifre) - maxi - mini     if suma == med:         print("special")     else: print("NU") </pre> | <pre> #include &lt;iostream&gt; #include &lt;algorithm&gt; using namespace std; int main() {     int n,maxi,mini,suma,suma_cif;     cout&lt;&lt;"n= ";     cin &gt;&gt; n;     if (n &gt;= 100 &amp;&amp; n &lt;= 999)     { int c1 = n / 100;       int c2 = (n / 10) % 10;       int c3 = n % 10;       maxi=max(max(c1,c2),c3);       mini=min(min(c1,c2),c3);       suma=mini+maxi;       suma_cif=c1+c2+c3-maxi-mini;       if ( suma==suma_cif)           cout &lt;&lt;"special";       else cout &lt;&lt; "NU";     }     return 0;} </pre> |

33. Scrieți un program în **Python/C++**, care citește trei numere reale pozitive. Programul verifică dacă cele trei numere citite pot fi laturile unui triunghi. În caz afirmativ, calculează și afișează pe ecran aria și perimetrul triunghiului. **De exemplu** pentru  $a=3$ ,  $b=4$ ,  $c=5$  se afișează **perimetrul=12** și **aria=6**.

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> import math # Citim laturile a = float(input("Latura a: ")) b = float(input("Latura b: ")) c = float(input("Latura c: ")) # Verificăm dacă pot forma un triunghi if a + b &gt; c and a + c &gt; b and b + c &gt; a:     perimetru = a + b + c     s = perimetru / 2     aria = math.sqrt(s * (s - a) * (s - b) * (s - c))     print(f"Perimetrul triunghiului este: {perimetru}")     print(f"Aria triunghiului este: {aria}") else: print("Triunghi invalid.") </pre> |

## Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
 double a, b, c, perimetru, s, aria;
 cout << "Introdu laturile triunghiului: ";
 cin >> a >> b >> c;
 if (a + b > c && a + c > b && b + c > a) {
 perimetru = a + b + c;
 s = perimetru / 2;
 aria=sqrt(s*(s-a)*(s-b)*(s-c));
 cout<<"Perimetrul triunghiului este: " << perimetru << endl;
 cout<<"Aria triunghiului este: " << aria << endl;
 }
 else cout<<"Triunghi invalid." << endl;
 return 0;
}
```

34. Într-un joc de criptografie, două cifre magice **x** și **y** (numere naturale nenule mai mici decât 10) sunt alese pentru a genera un cod numeric dublu: codul direct se obține așezând cifra **x** înaintea cifrei **y** (adică **xy**) și codul invers se obține așezând cifra **y** înaintea cifrei **x** (adică **yx**). Scrieți un program în **Python/C++**, care după generarea acestor două coduri, verifică dacă cel mai mare dintre cele două coduri este divizibil cu cel mai mic și afișeze ambele coduri pe aceeași linie. Dacă unul este divizibil cu celălalt se afișează mesajul Cod complet valid dacă nu, se afișează mesajul Cod parțial valid.

De exemplu, Pentru **x=4** și **y=2** se va afișa **42** și **24** **Cod parțial valid**, iar pentru **x=1** și **y=1** se va afișa **11** și **11** **Cod complet valid**

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                          | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x, y = map(int, input("Introduceți cifrele x și y (1-9): ").split())  if 0 &lt; x &lt; 10 and 0 &lt; y &lt; 10:     cod1 = x * 10 + y     cod2 = y * 10 + x      print(f"{cod1} {cod2}")     minim = min(cod1, cod2)     maxim = max(cod1, cod2)     if maxim % minim == 0:         print("Cod complet valid.")     else:         print("Cod parțial valid.")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int x, y, cod1, cod2;     cin &gt;&gt; x &gt;&gt; y;     if (x&gt;0 &amp;&amp; x&lt; 10 &amp;&amp; y&gt;0 &amp;&amp; y&lt;10) {         cod1=x*10+y; cod2=y*10+x;         cout&lt;&lt;cod1&lt;&lt;" "&lt;&lt;cod2&lt;&lt;endl;         int minim = min(cod1, cod2);         int maxim = max(cod1, cod2);         if (maxim % minim == 0)             cout&lt;&lt;"Cod complet valid."&lt;&lt; endl;         else             cout&lt;&lt;"Cod parțial valid."&lt;&lt; endl;     }     return 0; }</pre> |

35. Scrieți un program în **Python/C++**, care citește un număr natural nenul de două cifre, format doar din cifre pare (0, 2, 4, 6 sau 8). Programul transformă fiecare cifră a numărului după următoarele reguli: dacă cifra este divizibilă cu 4, se înlocuiește cu câtul împărțirii la 2, dacă cifra este mai mare decât 5, se înlocuiește cu modulul împărțirii la 3 iar celelalte cifre nu se modifică. Dacă numărul rezultat este identic cu cel inițial, se va afișa mesajul: fără modificări, altfel, se va afișa noul număr obținut.

**De exemplu**, pentru  $n=84$ , programul va afișa 42, iar pentru  $n=20$  programul va afișa fără modificări.

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("Introduceti un numar format doar din doua cifre pare: "))  if n &lt; 10 or n &gt; 99 or any(int(c) % 2 != 0 for c in str(n)):     print("Numarul trebuie sa aiba exact doua cifre pare.") else:     zeci = n // 10     unitati = n % 10      def transforma(c):         if c % 4 == 0:             return c // 2         elif c &gt; 5:             return c % 3         else:             return c      nou = transforma(zeci) * 10 + transforma(unitati)      if nou == n:         print("fara modificari")     else:         print(nou) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main(){     int n,zeci,unitati,nou_zeciș     int nou_unitati,n_nou;      cout &lt;&lt; "Introduceti un numar format din doua cifre pare: ";      cin&gt;&gt;n;     if((n &lt; 10    n &gt; 99)   ( zeci % 2 != 0    unitati % 2 != 0)){         cout &lt;&lt; "Numarul trebuie sa aiba exact doua cifre!"&lt;&lt; endl;              return 0;}     else{         zeci=n/10;    unitati=n%10;         if (zeci % 4 == 0)             nou_zeci= zeci/ 2;         else if (zeci &gt; 5)             nou_zeci=zeci % 3;         else nou_zeci=zeci;         if (unitati % 4 == 0)             nou_unitati= unitati/ 2;         else if (unitati &gt; 5)             nou_unitati=unitati % 3;         else nou_unitati=unitati;         n_nou=nou_zeci*10+ nou_unitati;         if (n_nou == n)             cout &lt;&lt; "fara modificari";         else cout &lt;&lt; n_nou ;     }     return 0; } </pre> |

36. Scrieți un program Python/C++ care să citească de la tastatură, de pe linii separate, două numere întregi **a** și **b** de exact două cifre, apoi să se calculeze și să afișeze, pe ecran, cel mai mic număr de două cifre care se poate forma cu cifrele unităților din fiecare număr. De exemplu, pentru  $a=32$  și  $b=21$ , va afișa: 12

| Limbajul Python                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a=int(input("a=")) b=int(input("b=")) if a//10 &lt; b//10:     print((a//10)*10+b//10) else:     print((b//10)*10+a//10) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a,b;     cout&lt;&lt;"a="; cin&gt;&gt;a;     cout&lt;&lt;"b="; cin&gt;&gt;b;     if(a/10&lt;b/10)         cout&lt;&lt;(a/10)*10+b/10;     else cout&lt;&lt;b/10*10+a/10;     return 0; } </pre> |

37. Scrieți un program Python/C++ care să citească de la tastatură un număr întreg  $n$  de exact trei cifre, apoi să se calculeze și să afișeze, pe ecran, cea mai mare cifră a numărului  $n$ . De exemplu, pentru  $n=352$  va afișa: 5

| Limbajul Python                                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input("n=")) maxim=n%10 if maxim&lt;(n//10)%10:     maxim=(n//10)%10 if maxim&lt;(n//100):     maxim=n//100 print(maxim) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,maxim;     cout&lt;&lt;"n=";     cin&gt;&gt;n;     maxim=n%10;     if(maxim&lt;n/10%10)         maxim=n/10%10;     if(maxim&lt;n/100)         maxim=n/100;     cout&lt;&lt;maxim;     return 0; } </pre> |

38. Scrieți un program Python/C++ care să citească de la tastatură trei numere reale pozitive  $a$ ,  $b$ ,  $c$ , apoi să afișeze, pe ecran, mesajul "Da" dacă cele trei numere pot fi laturile unui triunghi sau mesajul "Nu" în caz contrar. De exemplu, pentru  $a=7$ ,  $b=8$ ,  $c=9$  va afișa: Da

| Limbajul Python                                                                                                    | Limbajul C/C++                                                                            |
|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <pre> a=float(input("a=")) b=float(input("b=")) c=float(input("c=")) if a+b&gt;c and a+c&gt;b and b+c&gt;a: </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     float a,b,c; </pre> |

```

 print("Da")
else:
 print("Nu")

```

```

cout<<"a="; cin>>a;
cout<<"b="; cin>>b;
cout<<"c="; cin>>c;
if(a+b>c && a+c>b && b+c>a)
 cout<<"Da";
else cout<<"Nu";
return 0;
}

```

39. Scrieți un program **Python/C++** care citește un număr real **x** și afișează valoarea funcției date mai jos.

$$f(x) = \begin{cases} x-2, & \text{daca } x \leq 5 \\ 3 * (x-1), & \text{daca } x \in (5, 20) \\ x * (x-1), & \text{daca } x \geq 20 \end{cases}$$

| Limbajul Python                                                                                                            | Limbajul C/C++                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x=float(input("x=")) if x&lt;=5:     print(x-2) elif x&lt;20:     print(3*(x - 1)) else:     print(x*(x - 1)) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     float x;     cout&lt;&lt;"x="; cin&gt;&gt;x;     if(x&lt;=5) cout&lt;&lt;x-2;     else if(x&lt;20) cout&lt;&lt;3*(x-1);         else cout&lt;&lt;x*(x-1);     return 0; } </pre> |

40. Scrieți un program **Python/C++** care citește trei numere întregi **a, b, c**, și care afișează mesajul **"Da"** dacă valoarea lui **c** este divizor comun pentru celelalte două și mesajul **"Nu"** în caz contrar. De exemplu, pentru **a=4, b=8, c=2** va afișa: **Da**

| Limbajul Python                                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a=int(input("a=")) b=int(input("b=")) c=int(input("c=")) if a%c==0 and b%c==0:     print("Da") else:     print("Nu") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a,b,c;     cout&lt;&lt;"a="; cin&gt;&gt;a;     cout&lt;&lt;"b="; cin&gt;&gt;b;     cout&lt;&lt;"c="; cin&gt;&gt;c;     if(a%c==0 &amp;&amp; b%c==0)         cout&lt;&lt;"Da";     else cout&lt;&lt;"Nu";     return 0; } </pre> |

41. Scrieți un program Python/C++ care citește o literă **n** și care afișează litera următoare din alfabetul englez în cazul în care litera citită este o vocală iar în caz contrar afișează litera precedentă din alfabet. De exemplu, pentru **n='o'** va afișa: **p**

| Limbajul Python                                                                                                 | Limbajul C/C++                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=input("n=") if n in ['a','e','i','o','u']:     print(chr(ord(n)+1)) else:     print(chr(ord(n)-1))</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { char n;   cout&lt;&lt;"n="; cin&gt;&gt;n;   if(n=='a'    n=='e'    n=='i'        n=='o'    n=='u')     cout&lt;&lt;char(n+1);   else cout&lt;&lt;char(n-1);   return 0; }</pre> |

42. Scrieți un program Python/C++ care citește două numere naturale **a** și **b** (**a<b**) și care afișează câte pătrate perfecte sunt în intervalul [**a,b**].

| Limbajul Python                                                                                                                                          | Limbajul C/C++                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import math a=int(input("a=")) b=int(input("b="))  a1=math.sqrt(a) if a1*a1!=a:     a1=math.ceil(a1) b1=math.sqrt(b) print(int(b1)-int(a1)+1)</pre> | <pre>#include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std; int main() { int a,b,a1,b1;   cout&lt;&lt;"a="; cin&gt;&gt;a;   cout&lt;&lt;"b="; cin&gt;&gt;b;   a1=sqrt(a);   if(a1*a1!=a) a1++;   b1=sqrt(b);   cout&lt;&lt;b1-a1+1;   return 0; }</pre> |

43. Scrieți un program Python/C++ care citește două numere întregi nenule **a** și **b** (**a<b**) și care afișează mesajul "**Da**" dacă cele două numere au același semn și mesajul "**Nu**" în caz contrar. De exemplu, pentru **a=7**, **b=8** va afișa: **Da**

| Limbajul Python                                                                             | Limbajul C/C++                                                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a=int(input("a=")) b=int(input("b=")) if a*b&gt;0: print("Da") else: print("Nu")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { int a,b;   cout&lt;&lt;"a="; cin&gt;&gt;a;   cout&lt;&lt;"b="; cin&gt;&gt;b;   if(a*b&gt;0) cout&lt;&lt;"Da";   else cout&lt;&lt;"Nu";   return 0; }</pre> |

44. Scrieți un program Python/C++ care citește un număr natural  $a$  ce reprezintă anul unui eveniment. Se cere să se afișeze mesajul "Anul este bisect" dacă valoarea citită reprezintă un an bisect și mesajul "Nu este an bisect" în caz contrar.

#### Limbajul Python

```
a =int(input("a= "))
if a%400==0 or a%4==0 and a%100!=0 : print("Anul este bisect")
else: print("Nu este an bisect")
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{ int a;
 cout<<"a="; cin>>a;
 if ((a%400==0)|| (a%4==0 && a%100!=0)) cout<<"Anul este bisect";
 else cout<<"Nu este an bisect";
 return 0; }
```

45. Scrieți un program Python/C++ care citește patru numere naturale distincte  $a, b, c, d$  ce reprezintă capetele a două intervale  $[a, b]$  și  $[c, d]$  cu ( $a < b$  și  $c < d$ ). Se cere să se afișeze intervalul dat de intersecția celor două intervale citite sau mesajul "Multimea vida" dacă nu există interval de intersecție. De exemplu, pentru  $a=3$ ,  $b=8$ ,  $c=5$ ,  $d=9$  va afișa:  $[5, 8]$

#### Limbajul Python

```
a=int(input("a="))
b=int(input("b="))
c=int(input("c="))
d=int(input("d="))
inc=a
if inc<c:
 inc=c
fin=b
if fin>d:
 fin=d

if inc<=fin:
 print(f"[{inc},{fin}]")
else:
 print("Multimea vida")
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{ int a,b,c,d,inc,fin;
 cout<<"a="; cin>>a;
 cout<<"b="; cin>>b;
 cout<<"c="; cin>>c;
 cout<<"d="; cin>>d;
 inc=a;
 if(inc<c) inc=c;
 fin=b;
 if(fin>d) fin=d;
 if (inc<=fin)
 cout<<"["<<inc<<","<<fin<<"]";
 else cout<<"Multimea vida";
 return 0;
}
```

46. Scrieți un program care să citească de la tastatură un număr natural  $a$ , apoi să se afișeze, pe ecran, ultima cifră a numărului  $2^a$ . De exemplu, pentru  $a=31$  se va afișa: 8.

| Limbajul Python                                                                                                                                                        | Limbajul C/C++                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a = int(input("a= ")) if a==0:     print(1) else:     if a%4==0: print(6)     if a%4==1: print(2)     if a%4==2: print(4)     if a%4==3:         print(8) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a;     cout&lt;&lt;"a="; cin&gt;&gt;a;     if(a==0) cout&lt;&lt;1;     else     {         if(a%4==0) cout&lt;&lt;6;         if(a%4==1) cout&lt;&lt;2;         if(a%4==2) cout&lt;&lt;4;         if(a%4==3) cout&lt;&lt;8;     }     return 0; } </pre> |

47. Scrieți un program care să citească de la tastatură, două numere naturale nenule **a** și **b**, apoi un caracter **op**. Dacă caracterul **op** este 'a' se afișează media aritmetică a celor două numere naturale nenule. Dacă caracterul **op** este 'h' se afișează media armonică a celor două numere naturale nenule. Dacă caracterul **op** este 'g' se afișează media geometrică a celor două numere naturale nenule. Să se calculeze și să se afișeze, pe ecran, în funcție de opțiune, rezultatul mediei corespunătoare. De exemplu, pentru **a=31, b=17, op='h'** se va afișa: **21.9583**.

| Limbajul Python                                                                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> import math a = int(input("a= ")) b = int(input("b= ")) op= input("op= ") if op=='a':     print((a+b)/2) if op=='h':     print(2*a*b/(a+b)) if op=='g':     print(math.sqrt(a*b)) </pre> | <pre> #include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std; int main() {int a,b;     char op;     cout&lt;&lt;"a=";    cin&gt;&gt;a;     cout&lt;&lt;"b=";    cin&gt;&gt;b;     cout&lt;&lt;"op=";cin&gt;&gt;op;     if(op=='a') cout&lt;&lt;(a+b)/2.;     if(op=='h')         cout&lt;&lt;2.*a*b/(a+b);     if(op=='g')         cout&lt;&lt;1.*sqrt(a*b);     return 0; } </pre> |

48. Scrieți un program care să citească de la tastatură un caracter **c**. Dacă caracterul **c** este o literă mică a alfabetului englez, să se afișeze pe ecran, **Caracterul c este o literă mică!**. Dacă caracterul **c** este o literă mare a alfabetului englez, să se afișeze pe ecran, **Caracterul c este o literă mare!**. Dacă caracterul **c** **NU** este o literă mică și nici o literă mare a alfabetului englez, să se afișeze pe ecran, **Caracterul c NU este o literă!**. De exemplu, pentru **c='h'** se va afișa: **Caracterul h este o literă mică!**

## Limbajul Python

```
c= input("c= ")
if c>='a' and c<='z': print("Caracterul ",c," este o litera mica!")
else:
 if c>='A' and c<='Z':
 print("Caracterul ",c," este o litera mare!")
 else: print("Caracterul ",c," NU este o litera!")
```

## Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{ char c;

 cout<<"c=";<<cin>>c;

 if(islower(c))cout<<"Caracterul "<<c<<" este o litera mica!";
 else
 if(isupper(c)) cout<<"Caracterul "<<c<<" este o litera mare!";
 else cout<<"Caracterul "<<c<<" NU este o litera!";

 return 0; }
```

49. Scrieți un program care să citească de la tastatură o opțiune, în funcție de opțiune se vor citi datele de intrare, apoi se va calcula și afișa aria figurii geometrice. Dimensiunile figurilor geometrice sunt numere naturale nenule. Dacă caracterul **op** este 'p', se citește de la tastatură latura pătratului și se afișează aria acestuia. Dacă caracterul **op** este 'd', se citesc de la tastatură cele două dimensiuni, lungimea și lățimea dreptunghiului și se afișează aria acestuia. Dacă caracterul **op** este 'r' se citesc de la tastatură valorile celor două diagonale ale rombului și se afișează aria acestuia. Dacă caracterul **op** este diferit de caracterul 'p' sau 'd' sau 'r', atunci să se afișeze mesajul **NU este o figura geometrica!**. Să se calculeze și să se afișeze, pe ecran, în funcție de opțiune, rezultatul ariei corespunzătoare figurii geometrice alese. De exemplu, pentru **op='p'**, iar pentru latura pătratului se citește **l=5**, se va afișa: **Aria patratului= 25**.

## Limbajul Python

```
op= input("op= ")
if op=='p':
 l= int(input("l= "))
 print("Aria patratului= ",l*l)
else:
 if op=='d':
 L= int(input("L= "))
 l= int(input("l= "))
 print("Aria dreptunghiului =",L*l)
 else:
 if op=='r':
 d1= int(input("d1= "))
 d2= int(input("d2= "))
 print("Aria rombului = ",d1*d2/2)
 else: print("NU este o figura geometrica!")
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int a,b;
 cout<<"a= "; cin>>a;
 cout<<"b= "; cin>>b;
 cout<<a*100+b<<" ";<<b*100+a;
 return 0;
}
```

50. Să se scrie un program în Python/C++ care afișează pe ecran poziția punctului dat prin coordonatele carteziene  $A(x_A, y_A)$ . În funcție de poziția acestuia se va afișa un mesaj corespunzător.

De exemplu:

- pentru  $x_A=-2$  și  $y_A=-4$ , pe ecran se va afișa: **Punctul se află în cadranul III.**
- pentru  $x_A=0$  și  $y_A=-4$ , pe ecran se va afișa: **Punctul se află pe axa OY.**

## Limbajul Python

```
xA, yA = map(int, input().split())
Verificăm poziția punctului
if xA>0 and yA>0:
 print("Punctul se află în cadranul I")
if xA<0 and yA>0:
 print("Punctul se află în cadranul II")
if xA<0 and yA<0:
 print("Punctul se află în cadranul III")
if xA>0 and yA<0:
 print("Punctul se află în cadranul IV")
if xA!=0 and yA==0:
 print("Punctul se află pe axa OX")
if xA==0 and yA!=0:
 print("Punctul se află pe axa OY")
if xA==0 and yA==0:
 print("Punctul se află în centrul reperului cartezian")
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int xA, yA;
 cin >> xA >> yA;
 if (xA>0 && yA>0) cout<<"Punctul se află în cadranul I";
 if (xA<0 && yA>0) cout<<"Punctul se află în cadranul II";
 if (xA<0 && yA<0) cout<<"Punctul se află în cadranul III";
 if (xA>0 && yA<0) cout<<"Punctul se află în cadranul IV";
 if (xA!=0 && yA==0) cout<<"Punctul se află pe axa OX";
 if (xA==0 && yA!=0) cout<<"Punctul se află pe axa OY";
 if (xA==0 && yA==0)
 cout<<"Punctul se află în centrul reperului cartezian";
 return 0;
}
```

## IV. Structuri repetitive

### IV.1. Instrucțiunea repetitivă WHILE

1. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  și determină suma cifrelor lui  $n$ . Programul afișează pe ecran suma determinată. De exemplu, pentru  $n=1234$ , programul va afișa valoarea 10.

| Limbajul Python                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input("n= ")) S = 0 while n &gt; 0 :     S = S + n % 10     n = n // 10 print("Suma=", S)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   int n,S=0;     cout&lt;&lt;"n="; cin&gt;&gt;n;     while(n&gt;0)     {   S = S + n % 10;         n = n / 10;     }     cout&lt;&lt;"Suma="&lt;&lt;S;     return 0; }</pre> |

2. Într-o piscină sunt  $x$  litri de apă ( $x>0$ ). Fiind o vară extrem de călduroasă, după fiecare  $m$  minute, un sfert din cantitatea de apă din piscină se evaporă.

Scrieți un program **Python/C++** care să citească trei numere naturale  $x$ ,  $m$  și  $y$  și care să determine după câte minute vor rămâne în piscină cel mult  $y$  litri de apă.

**De exemplu**, pentru  $x=1000$ ,  $m=15$ ,  $y=505$  se va afișa 45 minute

| Limbajul Python                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x,m,y = map(int, input().split()) x=float(x) timp=0 while x&gt;y :     x=3*x/4     timp=timp+m print(timp)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {   float x;     int m,y,timp=0;     cin&gt;&gt;x&gt;&gt;m&gt;&gt;y;     while(x&gt;y)     {   x=3*x/4; timp=timp+m;}     cout&lt;&lt;timp;     return 0; }</pre> |

3. Scrieți un program **Python/C++** care citește succesiv, de pe linii separate, un șir de numere naturale până la citirea valorii 0 (care finalizează citirea). Programul determină și afișează pe ecran cel mai mare număr nenul din șirul citit împreună cu numărul său de apariții.

**De exemplu**, pentru șirul de valori: 5 12 7 31 7 21 31 3 8 0, se va afișa 31 2, cu semnificația: cel mai mare număr din șir 31 și apare de 2 ori.

| Limbajul Python                                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>nr=int(input()) maxnr=0 ap=0; while nr&gt;0:     if nr&gt;maxnr:         maxnr=nr         ap=1     else:         if nr==maxnr: ap=ap+1     nr=int(input()) print(maxnr,ap)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int nr,ap=0,maxnr=0;     cin&gt;&gt;nr;     while (nr&gt;0)     { if (nr&gt;maxnr)         {maxnr=nr;ap=1;}       else         if (nr==maxnr) ap++;       cin&gt;&gt;nr;     }     cout&lt;&lt;maxnr&lt;&lt;' '&lt;&lt;ap;     return 0; }</pre> |

4. Scrieți un program **Python/C++** citește de la tastatură un număr natural  $n$  și care să calculeze și să afișeze suma tuturor numerelor obținute prin permutări circulare la dreapta ale cifrelor lui  $n$ .

De exemplu, pentru  $n=132$ , programul calculează suma  $132+321+213$  și afișează 666.

| Limbajul Python                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) p=0 scif=0 while n&gt;0 :     scif=scif+n%10     n=n//10     p=p*10+1 S=scif*p print(S)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     long long S;     int n,scif=0,p=0;     cin&gt;&gt;n;     while(n&gt;0)         {scif=scif+n%10; n=n/10;           p=p*10+1;}     S=scif*p;     cout&lt;&lt;S;     return 0; }</pre> |

5. Scrieți un program **Python/C++** citește de la tastatură un număr natural  $n$  și care calculează suma tuturor numerelor distincte prefixe pentru  $n$ . Programul afișează pe ecran suma determinată. **De exemplu**, pentru  $n=21637$  programul afișează 24039 reprezentând suma prefixelor lui  $n$  ( $=2+21+216+2163+21637$ )

| Limbajul Python                                                                 | Limbajul C/C++                                                                                                                                                                               |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) Sp=0 while n&gt;0 :     Sp=Sp+n     n=n//10 print(Sp)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int Sp=0, n;     cin&gt;&gt;n;     while(n&gt;0)         { Sp=Sp+n; n=n/10; }     cout&lt;&lt;Sp;     return 0; }</pre> |

6. Scrieți un program **Python/C++** citește de la tastatură, de pe aceeași linie, un număr natural **n** și o cifră **k**. Programul calculează și afișează pe ecran produsul cifrelor lui **n**, diferite de cifra **k**. Dacă **n** nu conține cifre diferite de **k**, programul va afișa valoarea -1.

**De exemplu**, pentru **n=25632** și **k=2** programul va afișa 90, iar pentru **n=333**, **k=3** va afișa -1

| Limbajul Python                                                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n,k = map(int, input().split()) p=1 ok=0 while n&gt;0 :     c=n%10     n=n//10     if c!=k :         ok=1         p=p*c if ok==1 : print(p) else : print(-1)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n, p=1,k,ok=0,c;     cin&gt;&gt;n&gt;&gt;k;     while(n&gt;0)     {         c=n%10; n=n/10;         if(c!=k) { ok=1; p=p*c; }     }     if(ok==1) cout&lt;&lt;p;     else cout&lt;&lt;-1;     return 0; }</pre> |

7. Scrieți un program **Python/C++** citește de la tastatură, de pe aceeași linie, două numere naturale **x** și **y**, cu au același număr de cifre. Programul construiește numărul **z** format astfel: fiecare cifră a acestuia este egală cu partea întreagă a mediei aritmetice a cifrelor, cu același ordin de mărime, din numerele **x** și **y**. Programul va afișa pe ecran numărul construit. **De exemplu**, pentru **x=2514** și **y=5628**, programul va afișa 3516

| Limbajul Python                                                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x,y = map(int, input().split()) p=1 z=0 while x&gt;0 :     c = (x%10 + y%10)//2     x = x//10     y = y//10     z = z + c*p     p = p*10 print(z)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int x,y,z=0,p=1,c;     cin&gt;&gt;x&gt;&gt;y;     while(x&gt;0)     {         c=(x%10+y%10)/2;         x=x/10; y=y/10;         z=z+c*p; p=p*10; }     cout&lt;&lt;z;     return 0; }</pre> |

8. Scrieți un program **Python/C++** care citește succesiv, de pe linii separate, un șir de litere mici din alfabetul englez, separate prin câte un spațiu, până la citirea caracterului '#' (care finalizează citirea). Programul determină și afișează pe ecran prima literă din șirul citit împreună cu numărul său de apariții

**De exemplu**, pentru șirul de litere: **g a x g g a t b g #**, se va afișa **g 4**, cu semnificația: prima literă din șirul citit este **g** și apare de 4 ori.

| Limbajul Python                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> c=input() nrap=1 x=input() while x!='#':     if x==c : nrap=nrap+1     x=input() print(c, nrap) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     char c, x;     int nrap=1;     cin&gt;&gt;c;     cin&gt;&gt;x;     while (x!='#')     {         if (x==c) nrap++;         cin&gt;&gt;x;     }     cout&lt;&lt;c&lt;&lt;' '&lt;&lt;nrap;     return 0; } </pre> |

9. Scrieți un program **Python/C++** care citește succesiv, de pe linii separate, un șir de litere mici din alfabetul englez, separate prin câte un spațiu, până la citirea caracterului '#' (care finalizează citirea). Programul determină și afișează pe ecran câte vocale apar în șirul citit. Se consideră vocale literele **a, e, i, o, u**. **De exemplu**, pentru șirul de litere: **i a x g q a e b o #**, se va afișa **5**, cu semnificația: șirul citit conține 5 vocale.

| Limbajul Python                                                                                          | Limbajul C/C++                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x=input() nrap=0 while x!='#' :     if x in "aeiou" : nrap=nrap+1     x=input() print(nrap) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     char x;     int nrap=0;     cin&gt;&gt;x;     while (x!='#')     {         if (x=='a'    x=='e'    x=='i'                x=='o'    x=='u') nrap++;         cin&gt;&gt;x;     }     cout&lt;&lt;nrap;     return 0; } </pre> |

10. Scrieți un program **Python/C++** pentru a simula jocul „**Ghicește numărul**”, în care calculatorul generează un număr aleator în intervalul **[0,100]**. Utilizatorul încercă să ghicească numărul, tastând câte un număr din intervalul dat. Pentru fiecare număr tastat, pe ecran va apărea unul dintre mesajele: **Prea mare!**, **Prea mic** sau **Ai ghicit**. Jocul se termină atunci când apare pe ecran mesajul **Ai ghicit!** sau după maximum 7 încercări.

| Limbajul Python                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> import random s = 1 d = 100 k = 7 ok = 0 x = random.randint(0, 100) while k:     print("Număr încercări rămase =", k)     n = int(input("Scrie un număr = "))     k -= 1 </pre> |

```

if x == n:
 print("Ai ghicit!")
 ok = 1
 break
else:
 if n > x:
 d = n - 1
 print("Prea mare!")
 else:
 s = n + 1
 print("Prea mic!")
 if s == d:
 d += 1
if ok == 0: print("Ai pierdut!")

```

### Limbajul C/C++

```

#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{ int s=1, d=100,x,n,k=7,ok=0;
 x= rand()%101;
 while(k)
 { cout<<"Numar incercari ramase = "<<k<<endl;
 cout<<"scrie un numar=";
 cin>>n; k--;
 if(x==n) { cout<<"Ai ghicit!"; ok=1;break; }
 else {
 if(n>x)
 { d=n-1; cout<<"Prea mare!"<<endl; }
 else
 { s=n+1; cout<<"Prea mic!"<<endl; }
 if(s==d)d++;
 }
 }
 if(ok==0) cout<<"Ai pierdut!";
 return 0;
}

```

11. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^9$ ) și care calculează și afișează suma cifrelor impare ale lui  $n$ . Dacă  $n$  nu are cifre impare se afișează mesajul **fara impare**.

**Exemplu:** pentru  $n=514017$  se afișează 14

| Limbajul Python                                                                                                                                                                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) suma = 0 gasit = False if n == 0:     print("fara impare") else:     while n &gt; 0:         cifra = n % 10         if cifra % 2 == 1:             suma += cifra             gasit = True         n //= 10     if not gasit:         print("fara impare") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned n;     cin&gt;&gt;n;     int suma=0;     bool exista=false;     if(n==0)     {   cout&lt;&lt;"fara impare";         return 0;     } </pre> |

```

 n //= 10
 if gasit:
 print(suma)
 else:
 print("fara impare")

```

```

 }
 while(n>0)
 {
 int cifra=n%10;
 if(cifra%2==1)
 {
 suma+=cifra;
 exista=true;
 }
 n/=10;
 }
 if(exista) cout<<suma;
 else cout<<"fara impare";
 return 0;
}

```

12. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^9$ ) și care calculează și afișează numărul cifrelor pare ale lui  $n$ .

**Exemplu:** pentru  $n=20781$  se afișează 3

| Limbajul Python                                                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) count = 0 if n == 0:     count = 1 while n &gt; 0:     cifra = n % 10     if cifra % 2 == 0:         count += 1     n //= 10 print(count) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned n;     cin&gt;&gt;n;     int cnt=0;     if (n==0) cnt=1;     while(n&gt;0)     {         int cifra=n%10;         if(cifra%2==0) cnt++;         n/=10;     }     cout&lt;&lt;cnt;     return 0; } </pre> |

13. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^9$ ) și care calculează și afișează suma tuturor prefixelor pare ale lui  $n$ . Dacă nu există cifre pare în număr se afișează mesajul **fara pare**.

**Exemplu:** pentru  $n=234032$  se afișează 236608 ( $234032+2340+234+2=236608$ )

| Limbajul Python                                                                                                                                                                                                                      | Limbajul C/C++                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) temp = n nr_cifre = 0 while temp &gt; 0:     nr_cifre += 1     temp //= 10 putere = 10 ** (nr_cifre - 1) suma = 0 exista = False while putere &gt; 0:     prefix = n // putere     if prefix % 2 == 0: </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned n;     cin&gt;&gt;n;     unsigned temp=n;     int nr_cifre=0;     if(temp==0)         nr_cifre=1;     else         while(temp&gt;0) </pre> |

```

 suma += prefix
 exista = True
 putere //= 10
if exista:
 print(suma)
else:
 print("fara pare")

```

```

{
 nr_cifre++;
 temp/=10;
}
unsigned putere=1;
int i=1;
while(i<nr_cifre)
{
 putere*=10;
 i++;
}
unsigned suma=0;
bool exista=false;
while(putere>0)
{
 unsigned prefix=n/putere;
 if(prefix%2==0)
 {
 suma+=prefix;
 exista=true;
 }
 putere/=10;
}
if(exista)
 cout<<suma;
else
 cout<<"fara pare";
return 0;
}

```

14. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^9$ ) și, de pe același rând o cifră  $k$ . Programul afișează produsul cifrelor egale cu  $k$  sau mesajul **nu exista** dacă numărul  $n$  nu conține cifra  $k$ .

**Exemplu:** pentru  $n=234232$  și  $k=2$  atunci se afișează 8 iar pentru  $n=343$  și  $k=2$  atunci se afișează **nu exista**

| Limbajul Python                                                                                                                                                                                                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n, k = map(int, input().split()) produs = 1 gasit = False if n == 0 and k == 0:     produs = 0     gasit = True while n &gt; 0:     cifra = n % 10     if cifra == k:         produs *= k         gasit = True     n //= 10 if gasit:     print(produs) else:     print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned n;     int k;     cin&gt;&gt;n&gt;&gt;k;     unsigned temp=n;     int produs=1;     bool exista=false;     if (temp==0 &amp;&amp; k==0)     {         produs=0;         exista=true;     } </pre> |

```

while(temp>0)
{
 int cifra=temp%10;
 if (cifra==k)
 {
 produs*=k;
 exista=true;
 }
 temp/=10;
}
if(exista)
 cout<<produs;
else
 cout<<"nu exista";
return 0;
}

```

15. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $9 < n < 10^9$ ) și afișează mesajul **crescator** dacă cifrele sunt în ordine strict crescătoare, mesajul **descrescator** dacă cifrele sunt în ordine descrescătoare sau mesajul **nu** dacă cifrele nu sunt nici în ordine crescătoare nici în ordine descrescătoare.

**Exemplu:** pentru  $n=8410$  atunci se afișează **descrescator**

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) prev = n % 10 n //= 10 is_cresc = True is_descresc = True while n &gt; 0:     cifra = n % 10     if cifra &gt;= prev:         is_cresc = False     if cifra &lt;= prev:         is_descresc = False     prev = cifra     n //= 10 if is_cresc:     print("crescator") elif is_descresc:     print("descrescator") else:     print("nu") </pre> | <pre> #include &lt;iostream&gt; using namespace std;  int main() {     unsigned n;     cin&gt;&gt;n;     int prev=n%10;     n/=10;     bool crescator=true;     bool descrescator=true;     while(n&gt;0)     {         int cifra=n%10;         if(cifra&gt;=prev)             crescator=false;         if(cifra&lt;=prev)             descrescator=false;         prev=cifra;         n/=10;     }     if(crescator)         cout&lt;&lt;"crescator";     else if(descrescator)         cout&lt;&lt;"descrescator";     else         cout&lt;&lt;"nu";     return 0; } </pre> |

16. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, două numere naturale **a** și **b** ( $a, b < 10^9$ ). Programul afișează în ordine strict crescătoare cifrele comune celor două numere, separate printr-un spațiu, sau mesajul **nu exista** dacă **a** și **b** nu au cifre comune.

**Exemplu:** pentru **a=1020105** și **b=25802** atunci se afișează **0 2 5**

#### Limbajul Python

```
a, b = map(int, input().split())
def extrage_cifre(numar):
 cifre = []
 if numar == 0:
 cifre.append(0)
 while numar > 0:
 c = numar % 10
 if c not in cifre:
 cifre.append(c)
 numar //= 10
 return cifre
cifre_a = extrage_cifre(a)
cifre_b = extrage_cifre(b)
comune = []
for c in range(10):
 if c in cifre_a and c in cifre_b:
 comune.append(c)
if comune:
 for c in comune:
 print(c, end=' ')
else:
 print("nu exista")
```

#### Limbajul C++

```
#include <iostream>
using namespace std;
bool exista_cifra(unsigned numar, int cifra)
{
 if(numar==0)
 return cifra==0;
 while(numar>0)
 {
 if(numar%10==cifra)
 return true;
 numar/=10;
 }
 return false;
}

int main()
{
 unsigned a,b;
 cin>>a>>b;
 bool exista_comune=false;
 for(int c=0; c<=9; c++)
 if(exista_cifra(a,c) && exista_cifra(b,c))
 {
 cout<<c<<' ';
 exista_comune=true;
 }
 if (!exista_comune)
 cout<<"nu exista";
 return 0;
}
```

17. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, două numere naturale **a** și **b** ( $a, b < 10^9$ ). Programul afișează cea mai mică cifră comună celor două numere sau mesajul **nu exista** dacă **a** și **b** nu au cifre comune.

**Exemplu:** pentru **a=1927165** și **b=25802** atunci se afișează **2**

#### Limbajul Python

```
a, b = map(int, input().split())
cifre_a = set()
cifre_b = set()
temp = a
while temp > 0:
 cifre_a.add(temp % 10)
 temp //= 10
temp = b
while temp > 0:
 cifre_b.add(temp % 10)
 temp //= 10
comune = cifre_a.intersection(cifre_b)
if comune:
 print(min(comune))
else:
 print("nu exista")
```

#### Limbajul C++

```
#include <iostream>
using namespace std;
int main()
{
 unsigned a,b;
 cin>>a>>b;

 int min_comun=-1;
 for(int cifra=0; cifra<=9; cifra++)
 {
 unsigned temp=a;
 bool gasit_in_a=false;
 if(temp==0 && cifra==0) gasit_in_a=true;
 while(temp>0)
 {
 if(temp%10==cifra)
 { gasit_in_a=true; break; }
 temp/=10;
 }

 if (!gasit_in_a) continue;
 temp=b;
 bool gasit_in_b=false;
 if(temp==0 && cifra==0) gasit_in_b=true;
 while(temp>0)
 {
 if(temp%10==cifra) { gasit_in_b=true; break;}
 temp/=10;
 }
 if (gasit_in_b)
 { min_comun=cifra; break; }
 }
 if (min_comun==-1) cout<<"nu exista";
 else cout<<min_comun;
 return 0;
}
```

18. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^9$ ) și afișează numărul maxim de cifre alăturate identice în  $n$ .

**Exemplu:** pentru  $n=844111100$  atunci se afișează 4

#### Limbajul Python

```
n = int(input())
if n == 0: print(1)
else:
 max_repet = 1
 crt_repet = 1
 ult_cif = n % 10
 n //= 10
 while n > 0:
 cif = n % 10
 n //= 10
 if cif == ult_cif:
 crt_repet += 1
 if crt_repet > max_repet:
 max_repet = crt_repet
 else:
 crt_repet = 1
 ult_cif = cif
 print(max_repet)
```

#### Limbajul C++

```
#include <iostream>
using namespace std;
int main()
{
 unsigned n;
 cin >> n;
 if(n==0){ cout << 1; return 0; }
 int max_repet=1;
 int crt_repet=1;
 int ult_cif=n%10;
 n/=10;
 while(n>0)
 {
 int cif=n%10;
 n/=10;
 if(cif==ult_cif)
 {
 crt_repet++;
 if (crt_repet>max_repet)
 max_repet=crt_repet;
 }
 else { crt_repet=1; ult_cif=cif; }
 }
 cout << max_repet;
 return 0;
}
```

19. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^9$ ) și afișează cel mai mare subnumăr de trei cifre din  $n$ . Dacă  $n$  are mai puțin de trei cifre se afișează mesajul **cifre insuficiente**, iar dacă nu există un astfel de subnumăr se afișează mesajul **imposibil**.

**Exemplu:** pentru  $n=405176$  atunci se afișează 517

#### Limbajul Python

```
n = int(input())
if n < 100: print("cifre insuficiente")
else:

 max_sub = n%1000
 n=n//10
 while n > 99:
 if n%1000>max_sub:
 max_sub = n%1000
 n=n//10
 print(max_sub)
```

#### Limbajul C++

```
#include <iostream>
using namespace std;
int main()
{
 int n,max_sub;
 cin>>n;
 if(n<100) cout<<"cifre insuficiente";
 else
 {
 max_sub=n%1000;
 n=n/10;
 while(n>99)
 {
 if(n%1000>max_sub) max_sub=n%1000;
 n=n/10;
 }
 cout<<max_sub; }
 return 0;
}
```

20. Scrieți un program **Python/C++** care citește de la tastatură, de pe același rând, două numere naturale nenule  $a$  și  $b$  ( $a, b < 10^9$ ). Programul afișează cel mai mare divizor comun și cel mai mic multiplu comun al celor două numere, separate prin enter, ca în exemplu.

**Exemplu:** pentru  $a=18$  și  $b=24$  atunci se afișează

(18,24)=6

[18,24]=72

#### Limbajul Python

```
CMMDC folosind algoritmul lui Euclid
def calc_cmmdc(a, b):
 while b != 0:
 r = a % b
 a = b
 b = r
 return a
```

```

a, b = map(int, input().split())
CMMDC
cmmdc = calc_cmmdc(a, b)
CMMMNC
cmmmnc = a * b // cmmdc
Rezultatele in formatul cerut
print(f"({a},{b})={cmmdc}")
print(f"[{a},{b}]=cmmmnc")

```

### Limbajul C++

```

#include <iostream>
using namespace std;

// CMMDC folosind algoritmul lui Euclid
int calc_cmmdc(int a,int b)
{
 while(b!=0)
 { int r=a%b; a=b; b=r; }
 return a;
}

int main()
{
 int a,b;
 cin>>a>>b;

 int cmmdc=calc_cmmdc(a,b);
 // 1LL pentru a evita depasirea int
 long long cmmmnc=1LL*a*b/cmmdc;
 cout<<"("<<a<<" "<<b<<" ")= "<<cmmdc<<"\n";
 cout<<"["<<a<<" "<<b<<" "]= "<<cmmmnc;
 return 0;
}

```

21. Scrieți un program **Python/C++**, care citește de la tastatură un număr natural nenul  $n$ ,  $n < 20$  și afișează, pe ecran, al  $n$ -lea termen impar din șirul lui Fibonacci.

Șirul lui Fibonacci este generat astfel:  $\text{fib}(n)=1$ , dacă  $n=1$  sau  $n=2$ , iar  $\text{fib}(n)=\text{fib}(n-1)+\text{fib}(n-2)$ , dacă  $n>2$ .

**De exemplu**, pentru  $n=5$ , programul va afișa valoarea 13, deoarece al 5-lea număr impar din șirul lui Fibonacci: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... este 13.

### Limbajul Python

```

n = int(input("n="))
a = 1
b = 1
if n<=2: rez=1
else:
 while n-2 > 0 :
 c = a + b
 a = b
 b = c
 if c%2 != 0: n=n-1
 if n-2==0: rez=c

print(rez)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 int n,a,b,c,rez;
 cout<<"n="; cin>>n;
 a=1; b=1;
 if(n<=2) rez=1;
 else
 while(n-2>0)
 {
 c = a + b;
 a=b; b=c;
 if(c%2!=0) n=n-1;
 if(n-2==0) rez=c;
 }
 cout<<rez;
 return 0;
}

```

22. Scrieți un program **Python/C++**, care citește de la tastatură un număr natural  $x$  ( $x < 1000000$ ), apoi calculează și afișează, pe ecran, numărul cifrelor nenule din scrierea în baza 2 a numărului  $x$ .

**De exemplu**, pentru  $x=1024$ , se va afișa 1, iar pentru  $x=15$ , se va afișa 4

| Limbajul Python                                                                                    | Limbajul C/C++                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x = int(input("x=")) nrunu=0 while x&gt;0 :     nrunu=nrunu+x%2     x=x//2 print(nrunu)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int x,nrunu=0;     cin&gt;&gt;x;     while(x)         { nrunu+=x%2; x/=2;}     cout&lt;&lt;nrunu;     return 0; }</pre> |

23. Scrieți un program **Python/C++**, care citește succesiv, de pe linii separate, un șir de numere naturale, cu cel mult opt cifre fiecare, până la citirea valorii 0 (care finalizează citirea). Programul determină și afișează, pe ecran, câte numere din șirul citit au suma cifrelor reprezentării în baza 7 egală cu 3.

**De exemplu**, pentru șirul de valori: 3, 33, 51, 14, 1, 9, 0, se va afișa 3, cu semnificația: numerele 3, 51, 9 au suma cifrelor reprezentării în baza 7 egală cu 3, deoarece  $3=3 \cdot 7^0$ ,  $51=1 \cdot 7^2 + 0 \cdot 7^1 + 2 \cdot 7^0$  și  $9=1 \cdot 7^1 + 2 \cdot 7^0$ .

| Limbajul Python                                                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>x=int(input()) rez=0 while x&gt;0:     s=0     while x&gt;0:         s=s+x%7         x=x//7     if s==3: rez=rez+1     x=int(input()) print(rez)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int x,rez=0,s;     cin&gt;&gt;x;     while (x) {         s=0;         while(x) { s=s+x%7; x=x/7;}         if (s==3) rez++;         cin&gt;&gt;x; }     cout&lt;&lt;rez;     return 0; }</pre> |

24. Scrieți un program în **Python/C++**, care citește de la tastatură un număr natural nenul  $n$ , cu cel mult opt cifre, apoi calculează și afișează, pe ecran, cel mai mare exponent  $a$  din descompunerea  $n!=6^a \cdot b$ , unde  $n!=1 \cdot 2 \cdot \dots \cdot n$ .

**De exemplu**, dacă  $n=10$ , se va afișa: 4 deoarece  $6=3 \cdot 2$ , iar cifra 3 apare mai rar decât 2 în descompunere și  $10!=1 \cdot 2 \cdot (3 \cdot 1) \cdot 4 \cdot 5 \cdot (3 \cdot 2) \cdot 7 \cdot 8 \cdot (3 \cdot 3) \cdot 10$ ,

Adică este suficient să identificăm exponentul cel mai mare al lui 3 în  $10!$   
 $=3^4 \cdot 2^4 \cdot b=6^4 \cdot b$ , deci  $a=4$

**Explicația rezolvării:**  $n!$  conține  $[10/3]$  multipli nenuli ai lui 3 și  $[10/(3 \cdot 3)]$  multipli nenuli ai lui  $3 \cdot 3$ , deci rezultatul este  $[10/3] + [10/(3 \cdot 3)] = 3 + 1 = 4$ .

| Limbajul Python                                                                    | Limbajul C/C++                                                                                                                                                                                         |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) p,a=3,0 while n&gt;=p :     a+=n//p     p*=3  print(a) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,p=3,a=0;     cin&gt;&gt;n;     while (n&gt;=p)         {a=a+n/p;         p=p*3;}     cout&lt;&lt;a;     return 0; } </pre> |

25. Scrieți un program **Python/C++**, care citește de la tastatură un număr natural nenul  $n$  ( $n < 25$ ), reprezentând numărul de trepte pe care le are o scară. Știind că un roboțel poate să urce fie o treaptă, fie două trepte ale scării, programul afișează, pe ecran, numărul de posibilități distincte pe care le are roboțelul astfel încât să urce scara cu  $n$  trepte.

**De exemplu**, pentru  $n=4$  programul afișează 5 deoarece roboțelul are următoarele posibilități distincte de urcare a celor  $n$  trepte ale scării: 1111 (adică urcă de 4 ori câte o treaptă) sau 112 sau 121 sau 211 sau 22 (adică urcă de 2 ori câte 2 trepte).

| Limbajul Python                                                                                                                                                     | Limbajul C/C++                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) a,b=1,2 if n&lt;=2: rez=n else:     while n-2&gt;0:         c=a+b         a=b         b=c         n=n-1     if n-2==0: rez=c print(rez) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, a=1, b=2, c, rez;     cin&gt;&gt;n;     if(n&lt;=2) rez=n;     else         while (n-2&gt;0)             { c=a+b;a=b;b=c;n--;             if (n-2==0) rez=c;}     cout&lt;&lt;rez;     return 0; } </pre> |

26. Scrieți un program în **Python/C++** care citește de la tastatură două numere naturale:  $p$ , reprezentând lungimea primului salt al unei broscuțe, și  $m$ , reprezentând numărul total de salturi.

O broscuță curajoasă pornește într-o aventură sărind printr-un tărâm mlăștinos. Fiecare salt al broscuței este de două ori mai lung decât saltul anterior. Programul va calcula și afișa distanța totală parcursă de broscuță după  $m$  salturi.

Exemplu: dacă  $p = 2$  și  $m = 5$ , atunci broscuța sare:  $2 + 4 + 8 + 16 + 32 = 62$

Se va afișa: 62

| Limbajul Python                                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> p = int(input()) m = int(input()) total = 0 salt = p i = 1  while i &lt;= m:     total += salt     salt *= 2     i += 1  print(total) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int p, m;     cin &gt;&gt; p&gt;&gt;m;     int total = 0, salt = p, i = 1;     while (i &lt;= m) {         total += salt;         salt *= 2;         i++;     }     cout &lt;&lt; total;     return 0; } </pre> |

27. Să se scrie un program **Python/C++** care citește de la tastatură un număr natural  $n$  și afișează, pe ecran prima putere a lui 2 strict mai mare decât  $n$  și exponentul corespunzător. Exemplu: pentru  $n = 70$ , se va afișa  $2^7 = 128$ .

| Limbajul Python                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) putere = 1 exp = 0 while putere &lt;= n:     putere *= 2     exp += 1 print(putere) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     int putere = 1;     int exp = 0;     while (putere &lt;= n)         { putere *= 2; exp++; }     cout &lt;&lt; putere;     return 0; } </pre> |

28. Scrieți un program în **Python/C++** care citește de la tastatură un număr natural  $n$  și afișează, separate prin spațiu, primele  $n$  numere triunghiulare. Prietenul nostru, Darius, tocmai a învățat la școală despre numere triunghiulare. El a aflat că al  $k$ -lea număr triunghiular este suma primelor  $k$  numere naturale ( $1+2+3+\dots+k$ ). Fascinat de această proprietate, Darius vrea să afle care sunt primele  $n$  astfel de numere. Exemplu: Pentru  $n = 5$ , se va afișa: 1 3 6 10 15 (explicație: 1, 1+2, 1+2+3, 1+2+3+4, 1+2+3+4+5)

| Limbajul Python                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) i = 1 triunghiular = 0 while i &lt;= n:     triunghiular += i     print(triunghiular, end=' ')     i += 1 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,i = 1;     cin &gt;&gt; n;     int triunghiular = 0;     while (i &lt;= n) {         triunghiular += i;         cout &lt;&lt;triunghiular&lt;&lt;" ";         i++;     }     return 0;} </pre> |

29. Scrieți un program în **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n \geq 2$ ) și afișează un triunghi isoscel format din steluțe \*, cu baza și laturile pline, dar cu interiorul gol.

Exemplu : Pentru  $n=5$

```
 *
 * *
* *
* *

```

### Limbajul Python

```
n = int(input())
i = 1
while i < n:
 spaces = n - i
 print(" " * spaces, end="")
 if i == 1: print("*")
 else: print("*" + " " * (2 * i - 3) + "*")
 i += 1
Afișăm baza completă
print("*" * (2 * n - 1))
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;

int main() {
 int n, n=1;
 cin >> n;
 while (i < n) {
 int sp = 0;
 while (sp < n - i) { cout << " "; sp++; }
 if (i == 1) cout << "*";
 else {
 cout << "*";
 int inner = 0;
 while (inner < 2 * i - 3) { cout << " "; inner++; }
 cout << "*"; }
 cout << endl;
 i++;
 }
 int stars = 0;
 while (stars < 2 * n - 1) { cout << "*"; stars++; }
 cout << endl;
 return 0;
}
```

30. Scrieți un program **Python/C++**, care citește de la tastatură un șir de numere întregi de maxim 9 cifre până la întâlnirea lui 0 (0 nu face parte din șir). Să se afișeze inversul celui mai mic număr din șir.

**De exemplu:** pentru șirul 18 84 98 12 24 0 se va afișa 21

| Limbajul Python                                                                                                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x=int(input()) rez=0 while x&gt;0:     if x&lt;minim:         minim=x     x=int(input()) while minim&gt;0:     rez==rez*10+minim%10     minim==minim//10 print(rez) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, minim = 1000000000, rez=0;     cin&gt;&gt;x;     while (x) {         if (x &lt; minim) minim= n;         cin&gt;&gt;x; }     while (minim &gt; 0) {         rez = rez * 10 + minim % 10;         minim /= 10;         cout &lt;&lt; rez &lt;&lt; endl;         return 0;     } } </pre> |

31. Scrieți un program **Python/C++**, care citește de la tastatură un număr natural **n** ( $n < 1000000$ ). Să se construiască un nou număr format doar din **cifrele impare** ale lui **n**, păstrând ordinea inițială a acestora. Dacă nu există cifre impare, se va afișa mesajul: „Nu există cifre impare.” altfel se va afișa numărul construit.

**De exemplu,** pentru  $n=23456$ , se va afișa 35, iar

pentru  $n=2468$ , se va afișa „Nu există cifre impare.”

| Limbajul Python                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x = int(input("x="))  rez=0 p=1  nr_inv=0 while x&gt;0 :     cif=n%10     if cif%2 !=0:         nr_inv=nr_inv*10+cif         n=n//10  while nr_inv&gt;0:     rez==rez*10+nr_inv%10     nr_inv=nr_inv//10  if rez!=0: print("Numărul format din cifrele impare este:", rez) else: print("Nu există cifre impare in nr. initial.") </pre> |

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, rez = 0, cif;
 cin >> n;
 int p = 1, nr_inv = 0;
 while (n > 0) {
 cif = n % 10;
 if (cif % 2 != 0) nr_inv = nr_inv*10+ cif;
 n /= 10;
 }
 while (nr_inv > 0) {
 rez = rez*10+nr_inv% 10; nr_inv /= 10;
 }
 if (rez != 0)
 cout << "Numărul format din cifrele impare este: " << rez;
 else cout << "Nu există cifre impare în nr. initial.";
 return 0;
}
```

32. Scrieți un program **Python/C++**, care citește un număr natural  $n$ , cu cel mult opt cifre nenule fiecare. Programul determină dacă numărul citit este un număr mijlociu superior adică este format doar din cifre de 5 și 6 și va afișa pe ecran mesajul **mijlociu superior** dacă este mijlociu superior altfel mesajul **Nu**. De exemplu, pentru  $n=56565$ , se va afișa **mijlociu superior**, iar pentru  $n=2468$ , se va afișa **Nu**.

| Limbajul Python                                                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) ok = 1 while n &gt; 0:     cif = n % 10     if cif != 5 and cif != 6:         ok = 0         break     n //= 10 if ok:     print("mijlociu superior") else:     print("NU")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     long long n;     int ok=1,cif;     cin &gt;&gt; n;     while (n &gt; 0) {         cif = n % 10;         if (cif!= 5 &amp;&amp; cif != 6) {             ok = 0; break;         }         n /= 10;     }     if (ok)         cout &lt;&lt; "mijlociu superior."     else cout &lt;&lt; "NU";     return 0; }</pre> |

33. Scrieți un program în **Python/C++**, care citește de la tastatură un număr natural nenul  $n$ , cu cel mult opt cifre. Să se afișeze, pe ecran, cel mai mic număr prim mai mare sau egal decât  $n$ .

De exemplu, dacă  $n=14$ , se va afișa: 17

| Limbajul Python                                                                                                                                                                                                                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("n= ")) ok = 0 while True:     ok = 1     if n &lt; 2:         ok = 0     d = 2     while d * d &lt;= n and ok == 1:         if n % d == 0:             ok = 0         d += 1     if ok == 1:         break     n += 1 print("Cel mai mic numar prim mai mare sau egal cu n este:", n) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     cout &lt;&lt; "n= "; cin &gt;&gt; n;     int ok = 0;     while (true) {         ok = 1;         if (n &lt; 2) ok = 0;         int d = 2;         while(d * d &lt;=n &amp;&amp; ok== 1){             if (n % d == 0) ok = 0;             d++;         }         if (ok == 1) break;         n++;     }     cout &lt;&lt; "Cel mai mic numar prim mai mare sau egal cu n este: " &lt;&lt; n &lt;&lt; endl;     return 0; } </pre> |

34. Scrieți un program în **Python/C++**, care citește de la tastatură un număr natural nenul  $n$ , cu cel mult opt cifre. Să se afișeze, pe ecran, toate numerele naturale mai mici decât  $n$  care sunt divizibile cu 3.

**De exemplu**, pentru  $n=14$ , se va afișa: 3,6,9,12, iar pentru  $n=2$  se va afișa mesajul: nimic

| Limbajul Python                                                                                                                                                                                                                                               | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) if n &lt;= 1:     pass # Nu afișăm nimic else:     gasit = False     i = 3     while i &lt; n:         print(i, end=", ")         gasit = True         i += 3     if gasit:         print("\b\b ") # curăță virgula de la final </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     bool gasit = false;     i=3;     while(i&lt;n) {         cout &lt;&lt; i;         gasit = true;         if (i + 3 &lt; n) cout&lt;&lt;",";         i+=3;     }     if (!gasit) cout &lt;&lt; "nimic";     cout &lt;&lt; endl;     return 0; } </pre> |

35. Se dă un număr natural nenul  $n$ , reprezentând numărul de termeni ai unui șir de numere naturale, și cele  $n$  numere naturale. Să se afișeze pentru fiecare număr natural citit, cifra de control a acestuia. Scrieți un program Python/C++ care să calculeze și să afișeze cifra de control pentru fiecare termen al șirului. De exemplu, pentru  $n=5$  și șirul 31 3 17 2000 0, pe ecran se va afișa: 4 3 8 2 0.

| Limbajul Python                                                                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) i = 1 while i &lt;= n:     a = int(input())     if a==0:         print(a," ")     else:         if a%9==0:             print(9," ")         else:             print(a%9," ")     i += 1 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, a, i=1;     cin &gt;&gt;n;     while (i &lt;= n) {         cin&gt;&gt;a;         if(a==0)             cout&lt;&lt;a&lt;&lt;" ";         else if(a%9==0)             cout&lt;&lt;9&lt;&lt;" ";         else             cout&lt;&lt;a%9&lt;&lt;" ";         i+=1;}     return 0; } </pre> |

36. Se dă un număr natural nenul  $n$ , reprezentând numărul de termeni ai unui șir de numere naturale, și cele  $n$  numere naturale. Să se afișeze pentru fiecare număr natural citit, cea mai mare bază în care poate fi scris numărul. Scrieți un program Python/C++ care afișeze pe ecran, pentru fiecare număr natural citit, cea mai mare bază în care poate fi scris numărul. De exemplu, pentru  $n=5$  și șirul 31 83 17 2000 10, pe ecran se va afișa: 4 9 8 3 2.

| Limbajul Python                                                                                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) i = 1 while i &lt;= n:     a = int(input())     Max=0     while a&gt;0:         if a%10&gt;Max:             Max=a%10         a=a//10     print(Max+1," ")     i += 1 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, a, i=1;     cin &gt;&gt;n;     while (i &lt;= n){         cin&gt;&gt;a;         int Max=0;         while (a&gt;0){             if(a%10&gt;Max) Max=a%10;             a=a/10; }         cout&lt;&lt;Max+1&lt;&lt;" ";         i+=1;     }     return 0; } </pre> |

37. Se dă un număr natural nenul  $n$ , reprezentând numărul de termeni ai unui șir de numere naturale, și cele  $n$  numere naturale. Să se afișeze pentru fiecare număr natural citit, DA dacă numărul este fericit, respectiv NU altfel. Un număr este fericit dacă prima cifrăa numărului este strict mai mare decât restul cifrelor. Scrieți un program Python/C++ care afișeze pe ecran, pentru fiecare număr natural citit, DA/ NU dacă numărul este fericit/ nu este fericit. De exemplu, pentru  $n=5$  și șirul 31 83 17 2000 10, pe ecran se va afișa: DA DA NU DA DA.

| Limbajul Python                                                                                                                                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) i = 1 while i &lt;= n:     a = int(input())     Max=0     while a&gt;9:         if a%10&gt;Max:             Max=a%10         a=a//10     if a&gt;Max:         print("DA ")     else:         print("NU ")     i += 1 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt;n;     int i = 1, a;     while (i &lt;= n) {         cin&gt;&gt;a;         int Max=0;         while (a&gt;9)             { if(a%10&gt;Max)Max=a%10;a=a/10;}         if(a&gt;Max) cout&lt;&lt;"DA ";         else cout&lt;&lt;"NU ";         i+=1;     }     return 0; } </pre> |

38. Se dă un număr natural nenul  $n$ , reprezentând numărul de termeni ai unui șir de numere naturale, și cele  $n$  numere naturale cu exact patru cifre fiecare număr. Să se afișeze pentru fiecare număr natural citit, cel mai mare număr obținut dintre numărul format cu primele două cifre, respectiv numărul format cu ultimele două cifre ale numărului curent. Scrieți un program Python/C++ care afișeze pe ecran, cel mai mare număr obținut dintre numărul format cu primele două cifre, respectiv numărul format cu ultimele două cifre ale numărului curent. De exemplu, pentru  $n=5$  și șirul 3117 2383 2000 1003 2025, pe ecran se va afișa: 31 83 20 10 25.

| Limbajul Python                                                                                                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) i = 1 while i &lt;= n:     a = int(input())     if a%100&gt;a//100:         print(a%100," ")     else:         print(a//100," ")     i += 1 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, I, a;     cin &gt;&gt;n;     while (i &lt;= n) {         cin&gt;&gt;a;         if(a%100&gt;a/100)cout&lt;&lt;a%100&lt;&lt;" ";         else cout&lt;&lt;a/100&lt;&lt;" ";         i+=1;     }     return 0; } </pre> |

39. Se dă un număr natural nenul  $n$ , reprezentând numărul de termeni ai unui șir de numere naturale, și cele  $n$  numere naturale cu cel mult 13 cifre fiecare număr. Să se afișeze pentru fiecare număr natural citit, -1 dacă prima cifră a numărului  $nu$  este cifră primă, altfel să se afișeze numărul. Scrieți un program Python/C++ care afișeze pe ecran, pentru fiecare număr natural citit, -1 dacă prima cifră a numărului  $nu$  este cifră

primă, altfel să se afișeze numărul. De exemplu, pentru n=5 și șirul 3117 2383 2000 1003 8025, pe ecran se va afișa: 3117 2383 2000 -1 -1.

| Limbajul Python                                                                                                                                                                                                            | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) i = 1 while i &lt;= n:     a = int(input())     ca=a     while a&gt;9:         a=a//10     if a==2 or a==3 or a==5 or a==7:         print(ca," ")     else:         print(-1," ")     i += 1 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,i=1, a, ca;     cin &gt;&gt;n;     while (i &lt;= n) {         cin&gt;&gt;a; ca=a;         while (a&gt;9)a=a/10;         if(a==2    a==3   a==5               a==7)             cout&lt;&lt;ca&lt;&lt;" ";         else cout&lt;&lt;-1&lt;&lt;" ";         i+=1;     }     return 0;} </pre> |

## IV.2. Instrucțiunea repetitivă For

1. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 100$ ) și determină suma răsturnatelor tuturor numerelor naturale cel mult egale cu  $n$ . Programul afișează pe ecran suma determinată. De exemplu, pentru  $n=12$ , programul va afișa valoarea 78 ( $=1+2+3+\dots+9+1+11+21$ ).

| Limbajul Python                                                                                                                                                 | Limbajul C/C++                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) s = 0 for i in range(1, n + 1):     if i &lt; 10:  s += i     else:         x = (i % 10) * 10 + (i // 10)         s += x print(s) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {   int n,S=0,i,x;     cin&gt;&gt;n;     for(i=1;i&lt;=n;i++)         if(i&lt;10)S=S+i;         else             { x=i%10*10+i/10;               S=S+x;  }     cout&lt;&lt;S;     return 0; } </pre> |

2. Scrieți un program **Python/C++** care să citească două numere naturale  $a$  și  $b$  ( $10 < a < b < 10000$ ) și care afișeze pe ecran, în ordine crescătoare, separate prin câte un spațiu, toate numerele naturale din intervalul  $[a, b]$  care au cifra unităților strict mai mare decât cifra zecilor. **De exemplu**, pentru  $a=10$  și  $b=25$ , programul va afișa, pe o linie a ecranului: 12 13 14 15 16 17 18 19 23 24 25

| Limbajul Python                                                                                                               | Limbajul C/C++                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a,b = map(int, input().split()) s='' for c in range(a, b + 1):     if c//10%10&lt;c%10 : s=s+str(c)+' ' print(s) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {   int a,b,c;     cin&gt;&gt;a&gt;&gt;b;     for(c=a;c&lt;=b;c++)         if(c/10%10&lt;c%10)             cout&lt;&lt;c&lt;&lt;" ";     return 0; } </pre> |

3. Scrieți un program **Python/C++** care să afișeze, în ordine lexicografică, toate cuvintele palindrom formate din exact trei litere mici astfel: au prima literă și ultima literă consoane identice din mulțimea  $\{d, w\}$ , iar a doua literă este o vocală  $\{a, e, i, o, u\}$ .

**De exemplu**, primul cuvânt afișat este **dad**.

| Limbajul Python                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>lit13 = "dw" lit2 = "aeiou" for x in lit13 :     for y in lit2 :         print(x+y+x)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { char lit1[]="dw";   char lit2[]="aeiou";   int i,j;   for(i=0;i&lt;2;i++)       for(j=0;j&lt;5;j++)           {cout&lt;&lt;lit1[i]&lt;&lt;lit2[j];             cout&lt;&lt;lit1[i]&lt;&lt;endl;}   return 0; }</pre> |

4. Scrieți un program **Python/C++** care să afișeze toate cuvintele formate din exact trei litere astfel: prima literă este din mulțimea {a,b,c}, a doua literă este din mulțimea {x,y} iar ultima literă este din mulțimea {w,u}.

| Limbajul Python                                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>lit1 = ['a','b','c'] lit2=['x','y'] lit3=['w','u'] for x in lit1 :     for y in lit2 :         for z in lit3 :             print(x+y+z)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() { char lit1[]="abc";   char lit2[]="xy";   char lit3[]="wu";   int i,j,k;   for(i=0;i&lt;3;i++)       for(j=0;j&lt;2;j++)           for(k=0;k&lt;2;k++)               { cout&lt;&lt;lit1[i]&lt;&lt;lit2[j];                 cout&lt;&lt;lit3[k]&lt;&lt;endl;}   return 0; }</pre> |

5. Scrieți un program **Python/C++** care citește un număr natural  $n$  și apoi  $n$  numere naturale reprezentând temperaturile înregistrate într-un oraș timp de  $n$  zile. Programul determină temperaturile minimă și maximă din cele  $n$  zile precum și temperatura medie a temperaturilor înregistrate.

**De exemplu**, pentru  $n=7$  și temperaturile: 12, 16, 13, 5, 8, 23, 17 se vor afișa pe ecran valorile: 5 23 13.4286 , reprezentând temperaturile: minimă, maximă și medie.

| Limbajul Python                                                                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n,t = map(int, input().split()) t1=t2=t s=t for c in range(2, n + 1):     t=int(input())     s=s+t     if t&lt;t1 : t1=t     if t&gt;t2 : t2=t t3=s/n print(t1,t2,t3) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,t,t1,t2,i;     float t3,s;     cin&gt;&gt;n&gt;&gt;t;     t1=t2=t; s=t;     for(i=2;i&lt;=n;i++)     {         cin&gt;&gt;t;         if(t&lt;t1) t1=t;         if(t&gt;t2) t2=t;         s=s+t;     }     t3=s/n;     cout&lt;&lt;t1&lt;&lt;" "&lt;&lt;t2&lt;&lt;" "&lt;&lt;t3;     return 0; } </pre> |

6. Scrieți un program **Python/C++** care să citească o cifră nenulă **k** și să afișeze pe ecran tabla înmulțirii cu cifra respectivă în formatul din exemplu.

**De exemplu**, pentru **k=5**, programul va afișa: **1x5=5**

**2x5=10**

**3x5=15** , etc

| Limbajul Python                                                                   | Limbajul C/C++                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> k=int(input()) for c in range(1, 10):     print(c, 'x' ,k, '=' ,c*k) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {int k,i;   cin&gt;&gt;k;   for(i=1;i&lt;=9;i++)     cout&lt;&lt;k&lt;&lt;"x"&lt;&lt;i&lt;&lt;"="&lt;&lt;k*i&lt;&lt;endl;   return 0; } </pre> |

7. Scrieți un program **Python/C++** care să citească un număr natural nenul **n** (**n<101**)și care să afișeze pe ecran toate perechile distincte de numere naturale nenule (**a,b**) cu proprietatea că **a+b=n** și **a<b**. Programul va afișa perechile pe ecran.

**De exemplu**, pentru **n=7**, se vor afișa perechile: 1 6, 2 5, 3 4.

| Limbajul Python                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) for a in range(1, n//2+1):     for b in range(a+1, n):         if a+b==n:             print(a,b) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,a,b;     cin&gt;&gt;n;     for(a=1;a&lt;=n/2;a++)         for(b=a;b&lt;n;b++)             if(a+b==n)                 cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;endl;     return 0; } </pre> |

8. Scrieți un program **Python/C++** care să citească un număr natural nenul  $n$  ( $n < 101$ ) și care să afișeze pe ecran toate tripletele de numere naturale nenule  $(a, b, c)$  cu proprietatea că  $a \cdot b \cdot c = n$  și  $a < b < c$ . Programul va afișa tripletele pe ecran. Dacă nu există aceste triplete, programul va afișa mesajul **Nu există**.

De exemplu, pentru  $n=24$ , se vor afișa tripletele: 1 2 12, 1 3 8, 1 4 6, 2 3 4.

| Limbajul Python                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n=int(input()) for a in range(1, n//2+1):     for b in range(a+1, n):         for c in range(b+1,n):             if a*b*c==n:                 print(a,b,c)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {int n,a,b,c;   cin&gt;&gt;n;   for(a=1;a&lt;=n/2;a++)     for(b=a+1;b&lt;n;b++)       for(c=b+1;c&lt;=n;c++)         if(a*b*c==n)           cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;" "&lt;&lt;c&lt;&lt;endl;   return 0; }</pre> |

9. Scrieți un program **Python/C++** care să citească două numere naturale nenule  $n$  și  $k$  ( $n, k < 1000$ ). Programul determina și va afișa pe ecran rezultatul produsului  $n \cdot 10^k$ . De exemplu, pentru  $n=123$  și  $k=5$ , programul va afișa 12300000.

| Limbajul Python                                                                               | Limbajul C/C++                                                                                                                                                                    |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n,k=map(int,input().split()) s=str(n) for i in range(1, k+1):     s=s+'0' print(s)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {  int n,k,i;    cin&gt;&gt;n&gt;&gt;k; cout&lt;&lt;n;    for(i=1;i&lt;=k;i++) cout&lt;&lt;0;    return 0; }</pre> |

10. Definim un număr **3-prim** ca fiind un număr natural format doar din cifrele  $\{3, 5, 7\}$ . Scrieți un program **Python/C++** care să afișeze toate numerele **3-prime** formate din câte patru cifre.

| Limbajul Python                                                                                                                                                               | Limbajul C/C++                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>for i in range(3, 8,2):     for j in range(3, 8,2):         for k in range(3, 8,2):             for p in range(3, 8,2):                 print(i*1000+j*100+k*10+p)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {int i,j,k,p;   for(i=3;i&lt;=7;i+=2)     for(j=3;j&lt;=7;j+=2)       for(k=3;k&lt;=7;k+=2)         for(p=3;p&lt;=7;p+=2)           cout&lt;&lt;i*1000+j*100+k*10+p&lt;&lt;endl;   return 0; }</pre> |

11. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10^3$ ) și care calculează și afișează suma:  $1+1 \cdot 2+2 \cdot 3+3 \cdot 4+\dots+n \cdot (n+1)$ .

**Exemplu:** pentru  $n=5$  se afișează 71

| Limbajul Python                                                                                  | Limbajul C/C++                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input()) suma = 1 for i in range(1, n+1):     suma += i * (i + 1) print(suma)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n;     cin &gt;&gt; n;     int suma = 1;     for (int i = 1; i &lt;=n; i++)         suma += i * (i + 1);     cout&lt;&lt;suma;     return 0; }</pre> |

12. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 100$ ) și care calculează și afișează suma:  $1+1 \cdot 2+1 \cdot 2 \cdot 3+1 \cdot 2 \cdot 3 \cdot 4+\dots+1 \cdot 2 \cdot 3 \cdot \dots \cdot n \cdot (n+1)$ . **Exemplu:** pentru  $n=4$  se afișează 153

| Limbajul Python                                                                                                      | Limbajul C/C++                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import math n = int(input()) suma = 0 for k in range(1, n + 2):     suma += math.factorial(k) print(suma)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n;     cin&gt;&gt;n;     long long suma=0;     long long fact=1;     for (int i=1;i&lt;=n+1;++i)         {fact *= i;suma += fact;}     cout&lt;&lt;suma;     return 0; }</pre> |

13. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 10^6$ ) și care afișează mesajul **prim** dacă  $n$  este număr prim sau mesajul **neprim** în caz contrar.

**Exemplu:** pentru  $n=29$  se afișează **prim** iar pentru  $n=27$  se afișează **neprim**

| Limbajul Python                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import math n = int(input()) if n &lt;= 1: print("neprim") elif n == 2: print("prim") elif n % 2 == 0: print("neprim") else :     e_prim = True     for d in range(3, int(math.isqrt(n)) + 1, 2):         if n % d == 0:             e_prim = False             break     print("prim" if e_prim else "neprim")</pre> |

### Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
 int n;
 cin>>n;
 if (n<2)
 { cout<<"neprim"; return 0; }
 if (n==2)
 { cout<<"prim"; return 0; }
 if (n%2==0)
 { cout<<"neprim"; return 0;}
 bool e_prim=true;
 int limita=sqrt(n);
 for (int d=3;d<=limita;d += 2)
 if (n%d==0)
 { e_prim=false; break; }
 cout<<(e_prim?"prim":"neprim");
 return 0;
}
```

14. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $1 < n < 10^6$ ) și care afișează, separate printr-un spațiu, numerele prime mai mici sau egale cu  $n$ .

Exemplu: pentru  $n=30$  se afișează 2 3 5 7 11 13 17 19 23 29

### Limbajul Python

```
import math
def este_prim(numar):
 if numar < 2:
 return False
 if numar == 2:
 return True
 if numar % 2 == 0:
 return False
 for d in range(3, int(math.isqrt(numar)) + 1, 2):
 if numar % d == 0:
 return False
 return True

n = int(input())
for numar in range(2, n + 1):
 if este_prim(numar):
 print(numar, end=" ")
```

### Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
bool e_prim(int nr)
{
 if (nr < 2) return false;
 if (nr == 2) return true;
 if (nr % 2 == 0) return false;
```

```

 for (int d=3;d*d<=nr;d += 2)
 if (nr%d==0)
 return false;
 return true;
 }
 int main()
 {
 int n;
 cin >> n;
 for (int i=2;i<=n;++i)
 if (e_prim(i))
 cout<<i<<' ';
 return 0;
 }

```

15. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 100$ ) și apoi  $n$  numere naturale nenule cu cel mult 4 cifre, separate prin enter, și care afișează, separate printr-un spațiu, trei valori, astfel: valoarea maximă, valoarea minimă și media valorilor pare.

**Exemplu:** dacă pentru  $n$  se citește 5 iar valorile sunt 1 2 4 10 7 (separate prin enter) **se afișează** 10 1 5.33333 (numărul de zecimale este implicit)

#### Limbajul Python

```

n = int(input())
maxim = None
minim = None
suma_pare = 0
numar_pare = 0
for i in range(n):
 x = int(input())
 # inițializare minim și maxim la prima valoare
 if maxim is None or x > maxim:
 maxim = x
 if minim is None or x < minim:
 minim = x
 if x % 2 == 0:
 suma_pare += x
 numar_pare += 1
if numar_pare > 0:
 media_pare = suma_pare / numar_pare
else:
 media_pare = 0
print(maxim, minim, media_pare)

```

#### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 int n;
 cin >> n;
 int x;
 int maxim = 0, minim = 0;
 int suma_pare = 0, nr_pare = 0;
 for (int i = 0; i < n; ++i)
 {
 cin >> x;
 }

```

```

 if (i == 0) maxim = minim = x;
 else
 {
 if (x > maxim) maxim = x;
 if (x < minim) minim = x;
 }
 if (x % 2 == 0)
 { suma_pare += x; nr_pare++; }
}
double media=(nr_pare>0)?(double)suma_pare/nr_pare:0.0;
cout<<maxim<<' '<<minim<<' '<<media;
return 0;
}

```

16. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 100$ ) și apoi  $n$  numere naturale nenule cu cel mult 8 cifre, separate prin enter, și care afișează, separate printr-un spațiu, valorile care au exact același număr de cifre ca prima valoare dintre cele  $n$ .

**Exemplu:** dacă pentru  $n$  se citește 6 iar valorile sunt 212 3 1234 233 256 6789 (separate prin enter), se afișează 212 233 256

| Limbajul Python                                                                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) val1 = int(input()) cif1 = len(str(val1)) print(val1, end=' ') for _ in range(n - 1):     x = int(input())     if len(str(x)) == cif1:         print(x, end=' ') </pre> | <pre> #include &lt;iostream&gt; using namespace std; int nrCifre(int x) {     int cifre = 0;     while (x &gt; 0)     {         x /= 10; cifre++;     }     return cifre; } int main() {     int n;     cin &gt;&gt; n;     int x, val1 = 0, cif1=0;     for (int i=0; i&lt;n; ++i)     {         cin &gt;&gt; x;         if (i==0)         {             val1 = x;             cif1=nrCifre(val1);         }         if (nrCifre(x)==cif1)             cout&lt;&lt;x&lt;&lt;' ';     }     return 0; } </pre> |

17. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 100$ ) și apoi  $n$  numere naturale nenule cu cel mult 6 cifre, separate prin enter, și care afișează, separate printr-un spațiu, valorile care au exact același număr de divizori ca prima valoare dintre cele  $n$ .

**Exemplu:** dacă pentru  $n$  se citește 6 iar valorile sunt 10 34 7 80 8 14 (separate prin enter), se afișează 10 34 8 14

## Limbajul Python

```
def nr_divizori(x):
 if x == 1:
 return 1

 k = 0
 sqrt_x = int(x ** 0.5)
 for d in range(1, sqrt_x + 1):
 if x % d == 0:
 k += 1 if d == x // d else 2
 return k

n = int(input())
vall = int(input())
nr_div_vall = nr_divizori(vall)
print(f"{vall} ({nr_div_vall})", end=' ')
for _ in range(n - 1):
 x = int(input())
 nr_div_x = nr_divizori(x)
 if nr_div_x == nr_div_vall:
 print(x, end=' ')
```

## Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int nrDivizori(int x)
{
 if (x==1)
 return 1;
 int cnt=0;
 int rad=sqrt(x);
 for (int d=1;d<=rad;++d)
 if (x%d==0)
 cnt += (d==x/d)?1:2;
 return cnt;
}
int main()
{
 int n;
 cin>>n;
 int x, vall, nrDiv_vall;
 cin>>vall;
 nrDiv_vall=nrDivizori(vall);
 cout<<vall<<' ';
 for (int i=1;i<n;++i)
 {
 cin>>x;
 if (nrDivizori(x)==nrDiv_vall)
 cout<<x<<' ';
 }
 return 0;
}
```

18. Scrieți un program **Python/C++** care citește de la tastatură două numere naturale **a** și **b** ( $0 < a < b < 1000$ ), separate prin enter, și care afișează, separate printr-un spațiu, toate valorile din intervalul **[a,b]** care au suma cifrelor număr impar.

**Exemplu:** dacă **a=10, b=20**, atunci se afișează 10 12 14 16 18

| Limbajul Python                                                                                                                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>def suma_cifrelor(n):     return sum(int(c) for c in str(n))  a = int(input()) b = int(input()) for x in range(a, b + 1):     if suma_cifrelor(x) % 2 == 1:         print(x, end=' ')</pre> | <pre>#include &lt;iostream&gt; using namespace std; int sum_Cif(int x) {     int s=0;     while(x&gt;0)     { s += x%10; x /= 10; }     return s; } int main() {     int a, b;     cin&gt;&gt;a&gt;&gt;b;     for (int i=a;i&lt;=b;++i)         if (sum_Cif(i)%2==1)             cout&lt;&lt;i&lt;&lt;' ';     return 0; }</pre> |

19. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul **n** ( $n < 9$ ) și apoi, de pe rândul următor, separate prin spațiu, **n** cifre. Programul construiește și afișează numărul format doar cu cifrele pare dintre cele **n** cifre, în ordinea citirii. Dacă nu se citește nicio cifră pară se afișează mesajul **nu exista**. Se ignoră zerourile nesemnificative.

**Exemplu:** dacă **n=7**, iar cifrele sunt în această ordine 3 2 1 4 6 2 9, atunci se afișează 2462

| Limbajul Python                                                                                                                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input()) cifre = input().split() nr_format = "" for cifra in cifre:     if int(cifra) % 2 == 0:         nr_format += cifra if nr_format:     nr_int = int(nr_format)     print(nr_int) else: print("nu exista")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n;     cin&gt;&gt;n;     int cif;     bool cif1=false;     bool gasit=false;     int nr=0;     for(int i=0;i&lt;n;++i)     {         cin&gt;&gt;cif;         if(cif%2==0)         {             gasit=true;             if(!cif1 &amp;&amp; cif==0)                 continue;             cif1=true;             nr=nr*10+cif;         }     }     if(cif1    gasit) cout&lt;&lt;nr;     else cout&lt;&lt;"nu exista";     return 0; }</pre> |

20. Scrieți un program **Python/C++** care citește un număr natural nenul  $n$ . Programul îl scrie pe  $n$  ca sumă de două pătrate perfecte nenule, nu neapărat distincte, iar dacă  $n$  nu se poate descompune astfel, afișează mesajul **imposibil**. Dacă sunt mai multe soluții se vor afișa toate, pe rânduri diferite.

**Exemplu:** dacă  $n=13$  se afișează  $13=2^2+3^2$

#### Limbajul Python

```
import math
n = int(input())
gasit = False
for a in range(1, int(math.isqrt(n//2)) + 1):
 b_la_2 = n - a*a
 b = int(math.isqrt(b_la_2))
 if b*b == b_la_2 and b != 0:
 gasit = True
 print(f"{n}={a}^2+{b}^2",end='\n')
 #break
if not gasit:
 print("imposibil")
```

#### Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
 int n; bool gasit=false;
 cin>>n;
 for (int a=1;a*a<=n/2;++a)
 {
 int b2=n-a*a; int b=sqrt(b2);
 if (b>=1 && b*b==b2) {
 cout<<n<< '=' <<a<<"^2+"<<b<<"^2"<<'\n'; gasit=true; }
 }
 if (!gasit) cout<<"imposibil";
 return 0;
}
```

21. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $0 < n < 51$ ), apoi calculează și afișează, pe ecran, ultima cifră a sumei tuturor numerelor naturale impare, care aparțin intervalului  $[n, 2*n]$ . **De exemplu**, dacă  $n=4$ , programul va afișa valoarea 2 (în intervalul  $[4, 2*4]$  avem  $5+7=12$  iar ultima cifră este 2).

#### Limbajul Python

```
n = int(input())
m = (n<<1)
n = n+(n+1)%2
s = 0
for i in range(n, m + 1, 2):
 s=(s+(i%10))%10
print(s)
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n, m, s=0, i;
 cin>>n;
 m=(n<<1);
 for (i=n+(n+1)%2;i<=m;i=i+2)
 s=(s+(i%10))%10;
 cout<<s;
 return 0;
}
```

22. Scrieți un program **Python/C++** care citește două numere naturale  $a$  și  $b$  ( $0 < a < 10$ ,  $9 < b < 100$ ) și care afișează, pe ecran, în ordine descrescătoare, separate prin câte un spațiu, toate numerele naturale din intervalul  $[a, b]$ , care au cifra unităților o cifră pătrat perfect. **De exemplu**, pentru  $a=6$  și  $b=18$ , se va afișa: 14 11 10 9

| Limbajul Python                                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a,b = map(int, input().split()) s='' for i in range(b, a-1, -1):     if str(i%10) in "0149": s=s+str(i)+' ' print(s) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned int a,b,i,c;     cin&gt;&gt;a&gt;&gt;b;     for(i=b;i&gt;=a;i--)     {         c=i%10;         if(c&lt;=1    c==4    c==9)             cout&lt;&lt;i&lt;&lt;" ";     }     return 0; } </pre> |

23. Scrieți un program **Python/C++** care citește două numere naturale  $a$  și  $b$  ( $9 < a < b < 100$ ) și care afișează, pe ecran, în ordine crescătoare, separate prin câte un spațiu, toate numerele naturale din intervalul  $[a, b]$ , care au cifra zecilor o cifră primă. Dacă nu există niciun astfel de număr, pe ecran se afișează mesajul **nu exista**.

**De exemplu**, pentru  $a=12$  și  $b=19$ , se va afișa: nu exista

iar pentru  $a=12$  și  $b=23$ , se va afișa: 20 21 22 23

| Limbajul Python                                                                                                                                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a,b = map(int, input().split()) s='' ok=False for i in range(a, b+1):     if str(i//10) in "2357":         s=s+str(i)+' '         ok = True if ok:     print(s) else:     print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     unsigned int a,b,i,c,ok=0;     cin&gt;&gt;a&gt;&gt;b;     for(i=a;i&lt;=b;i++)     {         c=i/10;         if (c==2  c==3  c==5  c==7)             {cout&lt;&lt;i&lt;&lt;" "; ok=1;}     }     if(ok==0) cout&lt;&lt;"nu exista";     return 0; } </pre> |

24. Scrieți un program **Python/C++** care citește trei numere naturale  $a$ ,  $b$  și  $c$  ( $0 < a < b < 1000$ ,  $0 < c < 28$ ) și care afișează, pe ecran, în ordine descrescătoare, separate prin câte un spațiu, toate numerele naturale din intervalul  $[a, b]$ , care au suma cifrelor egală cu  $c$ . Dacă nu există niciun astfel de număr, pe ecran se afișează

mesajul nu exista.

**De exemplu**, pentru  $a=12$ ,  $b=15$ ,  $c=10$ , se va afișa: nu exista

iar pentru  $a=12$ ,  $b=23$ ,  $c=5$  se va afișa: 23 14

| Limbajul Python                                                                                                                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>a,b,c = map(int, input().split()) rez='' ok=False for i in range(b, a-1, -1):     s= i%10+i//10%10+i//100     if s==c:         rez=rez+str(i)+' '         ok=True if ok:     print(rez) else: print("nu exista")</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int a, b, c, i, s, ok=0;     cin&gt;&gt;a&gt;&gt;b&gt;&gt;c;     for(i=b;i&gt;=a;i--)         { s=i%10+i/10%10+i/100;           if(s==c)               {cout&lt;&lt;i&lt;&lt;" "; ok=1;}         }     if (ok==0)         cout&lt;&lt;"nu exista";     return 0; }</pre> |

25. Scrieți un program **Python/C++** care citește un număr natural  $n$  ( $1 < n < 100$ ) și apoi un șir de  $n$  numere întregi distincte, având cel mult 4 cifre fiecare, separate prin enter. Programul determină și afișează, pe ecran, diferența, în valoare absolută, dintre cele mai mici două numere pare distincte din șir. Dacă șirul de numere nu conține cel puțin două numere pare, programul afișează mesajul **nu exista**.

**De exemplu**, pentru  $n=7$  și numerele din șir sunt: -9, 8, 1, 4, -3, -4, -2 se va afișa pe ecran valoarea: 2, fiindcă cele mai mici numere pare sunt -4 și -2, iar  $|-4 - (-2)| = 2$

| Limbajul Python                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input("n=")) min1 = min2 = 10000 for i in range(n):     x = int(input("x="))     if x%2==0 and x&lt;min1:         min2=min1         min1=x     elif x%2==0 and x &lt; min2:         min2=x if min2==10000: print("nu exista") else:     dif= min2-min1     print(dif)</pre> |

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{ int n,min1=10000,min2=10000,i,x;
 cin>>n;
 for(i=1;i<=n;i++)
 { cin>>x;
 if(x%2==0 and x<min1) {min2=min1; min1=x;}
 else if(x%2==0 and x < min2)min2=x;
 }
 if(min2==10000)
 cout<<"nu exista";
 else cout<< min2-min1;
 return 0;
}
```

26. Scrieți un program **Python/C++** care citește o cifră nenulă **cif** și afișează, pe ecran, un triunghi cu cifra respectivă, în formatul din exemple. **De exemplu**, pentru **cif=4**, programul va afișa:

```
4
44
444
4444
```

iar pentru **cif=2**, programul va afișa:

```
2
22
```

| Limbajul Python                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>cif=int(input()) for i in range(1, cif+1):     rez=' '     for j in range(1, i+1):         rez=rez+str(cif)     print(rez)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {int cif,i,j;   cin&gt;&gt;cif;   for(i=1;i&lt;=cif;i++,cout&lt;&lt;'\\n')       for(j=1;j&lt;=i;j++)           cout&lt;&lt;cif;   return 0; }</pre> |

27. Scrieți un program **Python/C++** care citește un număr natural **n** ( $1 < n < 100$ ) și apoi un șir de **n** numere naturale, având cel puțin două și cel mult opt cifre fiecare, separate prin enter. Programul determină și afișează, pe ecran, câte numere din șir au proprietatea că pătratul primei cifre a unui număr este egal cu ultima cifră a celui număr.

**De exemplu**, pentru **n=6** și numerele din șir: 12, 224, 3009, 4902, 10, 10001 se va afișa pe ecran valoarea: 3, fiindcă  $2*2=4$ ,  $3*3=9$  și  $1*1=1$

| Limbajul Python                                                                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("n=")) cate=0 for i in range(n):     x = int(input("x="))     uc = x%10     while x&gt;9:         x=x//10     if uc==x*x: cate=cate+1 print(cate) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,cate=0,i,x,uc;     cin&gt;&gt;n;     for(i=1;i&lt;=n;i++)         {cin&gt;&gt;x; uc=x%10;         while (x&gt;9)x/=10;         if (x*x==uc) cate++;         }     cout&lt;&lt;cate;     return 0; } </pre> |

28. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $2 < n < 101$ ), apoi calculează și afișează, pe ecran, valoarea expresiei:  $3 \cdot 7 + 4 \cdot 9 + 5 \cdot 11 + \dots + n \cdot (2 \cdot n + 1)$ . **De exemplu**, pentru  $n=5$ , se va afișa: 112.

| Limbajul Python                                                                   | Limbajul C/C++                                                                                                                                                            |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) E=0 for i in range(3,n+1):     E=E+i*(2*i+1) print(E) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {int n,E=0,i;   cin&gt;&gt;n;   for(i=3;i&lt;=n;i++) E=E+i*(2*i+1);   cout&lt;&lt;E;   return 0; } </pre> |

29. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $1 < n < 1001$ ), apoi calculează și afișează, pe ecran, valoarea expresiei:

$$2 \cdot 3 + 3 \cdot 4 - 4 \cdot 5 + \dots + (-1)^n \cdot n \cdot (n+1)$$

**De exemplu**, pentru  $n=5$ , se va afișa: -16.

| Limbajul Python                                                                                                  | Limbajul C/C++                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) E=0 for i in range(2,n+1):     if i%2!=0: E=E-i*(i+1)     else: E=E+i*(i+1) print(E) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {int n,E=0,i;   cin&gt;&gt;n;   for(i=2;i&lt;=n;i++)       if(i%2)E=E-i*(i+1);       else E=E+i*(i+1);   cout&lt;&lt;E;   return 0; } </pre> |

30. Definim un număr **4-zigzag** ca fiind un număr natural format din 4 cifre, astfel: prima și a treia cifră sunt cifre impare din {1,3,5,7,9}, iar a doua și a patra cifră sunt cifre pare din {0,2,4,6,8}. Un exemplu de număr **4-zigzag** este 9450. Scrieți un program **Python/C++** care citește de la tastatură un număr natural **x**, având maxim 4 cifre, apoi afișează, pe ecran, toate numerele **4-zigzag**, în ordine descrescătoare, pe linii diferite, care sunt cel mult egale cu **x**.

| Limbajul Python                                                                                                                                                                                                                                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x=int(input()) ok=0 for a in range(9,0,-2):     for b in range(8,-1,-2):         for c in range(9,0,-2):             for d in range(8,-1,-2):                 e = a*1000+b*100+c*10+d                 if e&lt;x:                     print(e)                     ok=1 if ok==0: print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {int x,a,b,c,d,e,ok=0;   cin&gt;&gt;x;   for (a=9;a&gt;=1;a-=2)     for (b=8;b&gt;=0;b-=2)       for (c=9;c&gt;=1;c-=2)         for (d=8;d&gt;=0;d-=2)           {e=a*1000+b*100+c*10+d;             if (e&lt;x)               cout&lt;&lt;e&lt;&lt;endl,ok=1;           }   if (!ok) cout&lt;&lt;"nu exista";   return 0; } </pre> |

31. Scrieți un program **Python/C++** care citește de la tastatură un număr natural **n** reprezentând cifrele unui cod galactic. Nava Solaris trebuie să decoleze doar dacă suma cifrelor pare din cod este egală cu suma cifrelor impare. Programul va afișa pe ecran mesajul **DECOLARE**, dacă condiția este îndeplinită, sau **EROARE DE COD**, în caz contrar.

Exemplu: pentru **n = 263741** → cifre pare: 2, 6, 4 → sumă 12  
cifre impare: 3, 7, 1 → sumă 11 → se va afișa **EROARE DE COD**

| Limbajul Python                                                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = input() suma_pare = 0 suma_impere = 0 for cifra in n:     c = int(cifra)     if c % 2 == 0: suma_pare += c     else: suma_impere += c if suma_pare == suma_impere: print("DECOLARE") else: print("EROARE DE COD") </pre> |

## Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
 int n, copie, nr_cifre = 0;
 int cifre[20], suma_pare = 0, suma_impere = 0
 cout << "Introduceți codul galactic: ";
 cin >> n;
 copie = n;
 // Extragem cifrele și le stocăm într-un vector
 while (copie > 0) {
 cifre[nr_cifre] = copie % 10;
 copie /= 10;
 nr_cifre++;
 }
 // Parcurgem cifrele în ordine inversă
 for (int i = nr_cifre - 1; i >= 0; i--) {
 if (cifre[i] % 2 == 0) suma_pare += cifre[i];
 else suma_impere += cifre[i];
 }
 if (suma_pare == suma_impere)
 cout << "DECOLARE";
 else cout << "EROARE DE COD";
 return 0;
}
```

32. Scrieți un program Python/C++ care citește de la tastatură un număr  $n$  și construiește un model piramidal de apărare cu  $n$  nivele, afișând la fiecare nivel doar numere impare crescătoare, astfel:

Exemplu pentru  $n = 3$ :

```
1
3 5
7 9 11
```

## Limbajul Python

```
n = int(input("Introduceți numărul de niveluri: "))
numar = 1 # primul număr impar
for i in range(1, n + 1): # pentru fiecare nivel
 for j in range(i): # numere pe fiecare rând
 print(numar, end=' ')
 numar += 2 # trecem la următorul număr impar
 print() # trecem la rândul următor
```

## Limbajul C++

```
#include <iostream>
using namespace std;
int main() {
 int n, numar = 1;
 cout << "Introduceți numărul de niveluri: ";
 cin >> n;
 for (int i = 1; i <= n; i++) { // pentru fiecare nivel
 for (int j = 1; j <= i; j++) { // numerele de pe nivel
 cout << numar << " ";
 numar += 2; // trecem la următorul număr impar
 }
 cout << endl; // trecem la următorul rând
 }
 return 0;
}
```

33. Scrieți un program **Python/C++** care citește de la tastatură un șir de **n** caractere (cifre), și determină dacă cifrele parcurg un tipar crescător, descrescător sau sunt amestecate. Programul va afișa pe ecran unul dintre mesajele: „**crescător**”, „**descrescător**” sau „**neordonat**”.

Exemplu: pentru **n** = 5, șirul 1 2 3 4 5 → crescător  
pentru 5 4 3 2 1 → descrescător  
pentru 1 3 2 4 5 → neordonat

## Limbajul Python

```
n = int(input())
cifre = list(map(int, input().split()))
crescator = True
descrescator = True
for i in range(n - 1):
 if cifre[i] >= cifre[i + 1]:
 crescator = False
 if cifre[i] <= cifre[i + 1]:
 descrescator = False
if crescator:
 print("crescător")
elif descrescator:
 print("descrescător")
else:
 print("neordonat")
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;

int main() {
 int n, cifre[100];
 bool crescator = true, descrescator = true;
 cin >> n;
 cout << "Introduceți cifrele separate prin spațiu: ";
 for (int i = 0; i < n; i++) {
 cin >> cifre[i];
 }
 for (int i = 0; i < n - 1; i++) {
 if (cifre[i] >= cifre[i + 1])
 crescator = false;
 if (cifre[i] <= cifre[i + 1])
 descrescator = false;
 }

 if (crescator)
 cout << "crescător";
 else if (descrescator)
 cout << "descrescător";
 else
 cout << "neordonat";

 return 0;
}
```

34. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$ , reprezentând intensitatea scutului unei cetăți spațiale. Pentru a activa scutul complet, fiecare divizor propriu (exclusiv 1 și  $n$ ) al valorii  $n$  trebuie multiplicat cu poziția sa în ordinea crescătoare, iar produsul final al acestor valori trebuie afișat. Dacă  $n$  este prim, se va afișa **ACTIVARE IMPOSIBILĂ**.

## Limbajul Python

```
n = int(input())
divizori = []
produs = 1
Căutăm divizorii proprii (exclus 1 și n)
for i in range(2, n):
 if n % i == 0:
 divizori.append(i)
if len(divizori) == 0:
 print("ACTIVARE IMPOSIBILĂ")
else:
 for poz, val in enumerate(divizori, start=1):
 produs *= val * poz
 print(produs)
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, produs = 1, contor = 0;
 bool nrDivizori = false;
 cin >> n;
 for (int i = 2; i < n; i++) {
 if (n % i == 0) {
 contor++;
 produs *= i * contor;
 nrDivizori = true;
 }
 }
 if (!nrDivizori) cout << "ACTIVARE IMPOSIBILĂ";
 else cout << "Produsul activator: " << produs;
 return 0;
}
```

35. Scrieți un program Python/C++ care citește de la tastatură un număr natural  $n$ , reprezentând înălțimea turnului vrăjitorului. Pe fiecare nivel, vrăjitorul a așezat o cantitate de magie egală cu pătratul nivelului ( $1^2, 2^2, \dots, n^2$ ). Dacă nivelul este par, magia se scurge, iar energia devine negativă. Calculați energia totală acumulată în turn.  
Exemplu: pentru  $n = 4 \rightarrow 1^2 - 2^2 + 3^2 - 4^2 = 1 - 4 + 9 - 16 = -10$

## Limbajul Python

```
n = int(input())
energie = 0
for i in range(1, n + 1):
 if i % 2 == 0:
 energie -= i * i
 else:
 energie += i * i
print("Energia totală acumulată:", energie)
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, energie = 0;
 cin >> n;
 for (int i = 1; i <= n; i++) {
 if (i % 2 == 0) energie -= i * i;
 else energie += i * i;
 }
 cout << "Energia totală acumulată: " << energie;
 return 0;
}
```

36. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  și apoi  $n$  numere naturale, reprezentând simboluri magice gravate pe un inel antic. Inelul este activat doar dacă fiecare simbol este media aritmetică exactă (fără rest) a celor două simboluri vecine din stânga și din dreapta, cu așezare circulară (primul și ultimul simbol sunt vecini).

Dacă regula este respectată pentru toate simbolurile, se va afișa: **CERCUL ESTE STABIL** În caz contrar, se va afișa: **SIMBOLURILE NU SUNT ECHILIBRATE**.

Exemplu: Pentru  $n = 3$  și simboluri: 4 8 4

Vecini pt 8: 4 și 4  $\rightarrow (4+4)/2 = 4$

Vecini pt 4: 8 și 4  $\rightarrow (8+4)/2 = 6 \neq 4$

$\rightarrow$  SIMBOLURILE NU SUNT ECHILIBRATE

### Limbajul Python

```
n = int(input())
simboluri = list(map(int, input().split()))
stabil = True
for i in range(n):
 st = simboluri[(i - 1) % n]
 dr = simboluri[(i + 1) % n]
 media = (st + dr) // 2
 if (st + dr) % 2 != 0 or simboluri[i] != media:
 stabil = False
 break
if stabil:
 print("CERCUL ESTE STABIL")
else:
 print("SIMBOLURILE NU SUNT ECHILIBRATE")
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n;
 cin >> n;
 int a[1001];
 for (int i = 0; i < n; i++)
 cin >> a[i];
 bool stabil = true;
 for (int i = 0; i < n; i++) {
 int st = a[(i - 1 + n) % n];
 int dr = a[(i + 1) % n];
 if ((st + dr) % 2 != 0 || a[i] != (st + dr) / 2) {
 stabil = false;
 break;
 }
 }
 if (stabil) cout << "CERCUL ESTE STABIL";
 else cout << "SIMBOLURILE NU SUNT ECHILIBRATE";
 return 0;
}
```

37. Scrieți un program **Python/C++** care citește de la tastatură două numere naturale **n** și **m**. În Templul Cifrelor, doi ucenici trebuie să rostească fiecare câte un număr sacru. Legenda spune că dacă numerele rostite au măcar o cifră comună, se va deschide Poarta Cunoașterii.

Ajutați păstrătorul să decidă dacă cele două numere conțin cel puțin o cifră comună. Dacă da, afișați: **ACCES PERMIS** altfel, afișați: **ACCES RESPINS**

Exemplu: Pentru numerele: n = 247, m = 935 ⇒ **ACCES RESPINS** ,

#### Limbajul Python

```
n = int(input())
m = int(input())
vectori pentru frecvența cifrelor
cifre_n = [0] * 10
cifre_m = [0] * 10
extragem cifrele lui n
for cifra in str(n):
 cifre_n[int(cifra)] = 1
extragem cifrele lui m
for cifra in str(m):
 cifre_m[int(cifra)] = 1
verificăm dacă există cifră comună
acces = False
for i in range(10):
 if cifre_n[i] and cifre_m[i]:
 acces = True
 break
print("ACCES PERMIS" if acces else "ACCES RESPINS")
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, m;
 cin >> n >> m;
 int cifreN[10] = {0};
 int cifreM[10] = {0};
 // Extragem cifrele lui n
 for (int x = n; x > 0; x /= 10)
 cifreN[x % 10] = 1;
 // Extragem cifrele lui m
 for (int x = m; x > 0; x /= 10)
 cifreM[x % 10] = 1;
 // Verificăm dacă există o cifră comună
 bool comun = false;
 for (int i = 0; i < 10; i++) {
 if (cifreN[i] && cifreM[i])
 { comun = true; break; }
 }
 if (comun) cout << "ACCES PERMIS";
 else cout << "ACCES RESPINS";
 return 0;
}
```

38. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $0 < n < 51$ ), apoi calculează și afișează prima cifră a produsului tuturor numerelor pare din intervalul  $[n, 2 \cdot n]$ . Dacă nu există numere pare în acest interval, se va afișa mesajul: „Nu există numere pare în interval.”

**De exemplu**, pentru  $n = 5$ , intervalul este  $[5, 10]$ , numerele pare sunt 6, 8, 10, iar produsul este  $6 \cdot 8 \cdot 10 = 480$ . Prima cifră este 4.

| Limbajul Python                                                                                                                                                                                                                                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("Introduceți un nr n: ")) produs = 1 are_pare = False for i in range(n, 2 * n + 1):     if i % 2 == 0:         produs *= i         are_pare = True if are_pare:     p_cif = int(str(produs)[0])     print("Prima cifră este:", p_cif) else:     print("Nu există numere pare") </pre> | <pre> #include &lt;iostream&gt; using namespace std;  int main() {     int n;     long long prod = 1;     bool are_pare = false;     cout &lt;&lt; "Introduceti un nr n: ";     cin &gt;&gt; n;     for(int i = n; i &lt;= 2 * n; i++)     {         if (i % 2 == 0)         { prod*= i;           are_pare = true;         }     }     if (are_pare) {         while (prod &gt;= 10)             prod /= 10;         cout&lt;&lt;"Prima cif e: "&lt;&lt; prod;     }     else         cout &lt;&lt; "Nu exista numere pare";     return 0; } </pre> |

39. Scrieți un program în **Python/C++** care citește două numere naturale  $x$  și  $y$  ( $5 < x < 20$ ,  $20 < y < 100$ ), și afișează, în ordine crescătoare, toate numerele pare din intervalul  $[x, y]$  al căror număr de cifre este egal cu suma cifrelor lor modulo 2. Dacă nu există nici un număr se va afișa mesajul **Nu există**

**De exemplu**, pentru  $x = 12$ ,  $y = 30$ , se va afișa mesajul **Nu există**

| Limbajul Python                                                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> x = int(input("Introduceți x: ")) y = int(input("Introduceți y: ")) exist=0 for i in range(x, y + 1):     if i % 2 == 0:         suma=sum(int(c) for c in str(i)) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int x, y, cifre, suma, exist=0;     cout &lt;&lt; "Introduceti x: ";     cin &gt;&gt; x;     cout &lt;&lt; "Introduceti y: ";     cin &gt;&gt; y; </pre> |

```

nr_cifre = len(str(i))
if suma % 2 == nr_cifre:
 print(i, end=' ')
 exista = 1
if not exista:
 print("Nu există")

```

```

for (int i = x; i <= y; i++) {
 if (i % 2 == 0) {
 int suma = 0; cop = i;
 while(cop) {
 suma += n % 10; cop /= 10; }
 cifre = 0; cop = i;
 while(cop) {
 cifre++;
 cop /= 10; }
 if (suma % 2 == cifre) {
 cout << i << " ";
 exist = 1; }
 }
}
if (!exista)
 cout << "Nu există";
return 0;
}

```

40. Scrieți un program în **Python/C++** care citește două numere naturale  $m$  și  $n$  ( $10 \leq m < n \leq 99$ ), și care afișează, pe ecran, în ordine crescătoare, separate prin câte un spațiu, toate numerele din intervalul  $[m, n]$  care **au produsul dintre cifra zecilor și cifra unităților impar**. Dacă nu există niciun astfel de număr, pe ecran se afișează mesajul **nu exista**.

De exemplu, pentru  $a=11$  și  $b=20$ , se va afișa: 11 13 15 17 19

iar pentru  $a=12$  și  $b=23$ , se va afișa: 20 21 22 23

| Limbajul Python                                                                                                                                                                                                                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> m = int(input("Introduceți m: ")) n = int(input("Introduceți n: ")) exista = False for numar in range(m, n + 1):     zeci = numar // 10     unitati = numar % 10     produs = zeci * unitati     if produs % 2 == 1:         print(numar, end=' ')         exista = True if not exista:     print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int m, n, zeci, unitati;     bool exista = false;     cout &lt;&lt; "Introduceți m :"; cin &gt;&gt; m;     cout &lt;&lt; "Introduceți n :"; cin &gt;&gt; n;     for (int i = m; i &lt;= n; i++) {         zeci = i / 10;         unitati = i % 10;         if ((zeci * unitati) % 2 == 1)         {             cout &lt;&lt; i &lt;&lt; " ";             exista = true;         }     }     if (!exista) cout &lt;&lt; "nu exista";     return 0; } </pre> |

41. Scrieți un program în **Python/C++** care citește trei numere naturale  $p, q$  și  $s$  ( $9 < p < q < 999$ ,  $1 \leq s \leq 9$ ) și afișează, pe ecran, în ordine **crescătoare**, separate prin câte un spațiu, toate numerele din intervalul  $[p, q]$  care conțin **exact o singură cifră egală cu  $s$** . Dacă nu există niciun astfel de număr, pe ecran se afișează mesajul **nu exista**.

**De exemplu**, pentru  $p=12, q=25, s=2$ , se va afișa: **nu exista**

iar pentru  $p=12, q=25, s=2$  se va afișa: **12 23 25**

| Limbajul Python                                                                                                                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> p,q,s = map(int, input().split()) exist=False for i in range(p, q+1):     aparitii =str(numar).count(str(s))     if aparitii == 1:         print(numar, end=' ')         exist = True if not exist:     print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() { int p, q, s, i, s, exist=0;   cin&gt;&gt;p&gt;&gt;q&gt;&gt;s;   for(i=p;i&lt;=q;i++)       { int count = 0, numar=i;         while (numar &gt; 0) {             if (numar % 10 == s)                 count++;             numar /= 10;}         if(count==1)             {cout&lt;&lt;i&lt;&lt;" "; exist=1;}         }   if (exist==0)       cout&lt;&lt;"nu exista";   return 0; } </pre> |

42. Scrieți un program **Python/C++** care citește un număr natural  $n$  ( $2 \leq n \leq 99$ ) și apoi un șir de  $n$  numere întregi distincte, fiecare cu cel mult 4 cifre. Programul va afișa, pe ecran, cea mai mare valoare care se poate obține prin înmulțirea a două valori diferite dintre cele citite. Dacă în șirul de numere nu există cel puțin două numere diferite, se va afișa mesajul: **nu exista**

**De exemplu**, pentru  $n=4$  și numerele din șir sunt:  $-5, 3, 7, -2$ , se va afișa pe ecran valoarea: **21** ( $3 \cdot 7$ )

| Limbajul Python                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("Introduceți n: ")) nr1 = int(input("Introduceți primul număr: ")) nr2 = int(input("Introduceți al doilea număr: ")) max1 = nr1 max2 = nr2 if max2 &gt; max1:     max1, max2 = max2, max1 exista_diferite = (nr1 != nr2) for i in range(3, n + 1):     if x != max1 and x != max2: </pre> |

```

 exista_diferite = True
 if x > max1:
 max2 = max1
 max1 = x
 elif x > max2: max2 = x
 if exista_diferite:
 print("Cea mai mare valoare este:", max1 * max2)
 else: print("nu exista")

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main() {
 int n, x;
 cout << "Introduceti n: ";cin >> n;
 int nr1, nr2;
 cout << "Introduceti primul numar: "; cin >> nr1;
 cout << "Introduceti al doilea numar: ";cin >> nr2;
 int max1 = nr1, max2 = nr2;
 if (max2 > max1) { int aux = max1; max1 = max2; max2 = aux; }
 for (int i = 3; i <= n; i++) {
 cin >> x;
 if (x != max1 && x != max2) {
 if (x > max1) {max2 = max1; max1 = x;}
 else if (x > max2) { max2 = x;}
 }
 }
 if (max1 != max2)
 cout << "Cea mai mare valoare este: " << max1 * max2;
 else cout << "nu exista";
 return 0;
}

```

43. Scrieți un program în **Python/C++** care citește o cifră nenulă  $c$  ( $1 \leq c \leq 9$ ) și afișează un triunghi numeric inversat, în care pe prima linie apar  $c$  cifre  $c$ , pe a doua linie  $c-1$  cifre  $c$ , și așa mai departe, până la o singură cifră  $c$ .

**De exemplu**, pentru  $c=3$ , programul va afișa:

```

333
33
3

```

| Limbajul Python                                                                                                                     | Limbajul C/C++                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> cif=int(input()) for i in range(c, 0,-1):     rez=''     for j in range(0, i):         rez=rez+str(cif)     print(rez) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {int cif,i,j;   cin&gt;&gt;c;   for(i=c;i&gt;=1;i--,cout&lt;&lt;'\\n')     for(j=0;j&lt;i;j++)  cout&lt;&lt;c;   return 0; } </pre> |

44. Scrieți un program **Python/C++** care citește un număr natural  $n$  ( $1 < n < 100$ ) și apoi un șir de  $n$  numere naturale, având cel puțin două și cel mult patru cifre fiecare, separate prin enter. Programul determină și afișează, pe ecran, cel mai mic număr care are prima cifră egală cu ultima cifră. Dacă nu există niciun astfel de număr, să se afișeze mesajul: „nu exista”.

**De exemplu**, pentru  $n=5$  și numerele din șir: 1231, 2442, 458, 707, 3423, se va afișa pe ecran valoarea: 707

| Limbajul Python                                                                                                                                                                                                                                           | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("Introduceți n")) mini=0 for i in range(1,n):     x = int(input("x="))     uc = x%10     while x&gt;9:         x=x//10     if uc==x: if minim==0 or x&lt;mini: mini=x if mini!=0:     print(mini) else:     print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() { int n, x, minim = -1; cin &gt;&gt; n; for (int i = 0; i &lt; n; i++) {     cin &gt;&gt; x;     int ultima = x % 10;     while (x &gt;= 10) x /= 10;     int prima = x;     if (prima == ultima) {         if (minim == -1    x &lt; minim)             minim = x; } } if (minim != -1) cout &lt;&lt; minim; else cout &lt;&lt; "nu exista"; return 0; } </pre> |

45. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $2 < n < 101$ ). Programul determină și afișează, pe ecran, **suma pătratelor primelor  $n$  numere impare**.

**De exemplu**, pentru  $n=3$ , se va afișa: 35 (1+9+25) .

| Limbajul Python                                                                                                                                                            | Limbajul C/C++                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("Introduceți n: ")) suma = 0 for i in range(1,n):     numar_impar = 2 * i + 1     suma += numar_impar ** 2 print("Suma pătratelor este:", suma) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() { int n, sumanr_impar; cout &lt;&lt; "Introduceți n: "; cin &gt;&gt; n; suma = 0; for (int i = 0; i &lt; n; i++) {     int nr_impar = 2 * i + 1;     suma +=nr_impar* nr_impar; } cout &lt;&lt; "Suma pătratelor este: " &lt;&lt; suma; return 0; } </pre> |

46. Scrieți un program **Python/C++** care citește un număr natural  $n$  ( $1 < n < 100$ ) și apoi un șir de  $n$  numere naturale, având cel puțin două și cel mult patru cifre fiecare, separate prin enter. Programul determină și afișează, pe ecran, suma celor care au exact trei cifre.

**De exemplu**, pentru  $n=5$  și numerele din șir: 45 123 999 1003 300, se va afișa: 1422.

| Limbajul Python                                                                                                                                                                 | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input("Introduceți nr n: ")) suma = 0 for i in range(0,n):     x = int(input())     if 100 &lt;= x &lt;= 999:         suma += x print("Suma nr este:", suma) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, x, suma = 0;     cout &lt;&lt; "Introduceți numărul n: ";     cin &gt;&gt; n;     for (int i = 0; i &lt; n; i++) {         cin &gt;&gt; x;         if (x &gt;= 100 &amp;&amp; x &lt;= 999)             suma += x;     }     cout &lt;&lt; "Suma nr este: " &lt;&lt; suma;     return 0; } </pre> |

47. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  și  $d$  și apoi un șir de  $n$  numere naturale, având cel puțin două și cel mult patru cifre fiecare. Programul determină și afișează, pe ecran, **produsul** tuturor numerelor **pare** dintre cele  $n$  citite și **câte** dintre aceste numere **au ultima cifră egală cu  $d$** .

**De exemplu**, pentru  $n=6$  și  $d=4$  și numerele din șir: 12 23 34 40 15 4 se va afișa: 65280 (  $12 \cdot 34 \cdot 40 \cdot 4$  ) și: Nr pare cu cifra unitatilor 4 = 2

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input("Introduceți n: ")) d = int(input("Introduceți d: ")) prod_pare = 1 count_d = 0 are_pare = False for i in range(0,n):     x = int(input())     if x % 2 == 0:         prod_pare *= x         are_pare = True     if x % 10 == d:         count_d += 1 if are_pare: print("Produsul nr pare este:", prod_pare) else: print("Nu există numere pare.") print(f"Nr pare cu cifra unitatilor {d}: {count_d}") </pre> |

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, d, x, are_pare=0, prod=1, count=0;
 cout << "Introduceți n și d: ";
 cin >> n >> d;
 for (int i = 0; i < n; i++) {
 cin >> x;
 if (x % 2 == 0) {
 prod*= x; are_pare = 1;
 if (x % 10 == d) count++;
 }
 }
 if (are_pare)
 cout << "Produsul numerelor pare este: " << prod<< endl;
 else cout << "Nu există numere pare." << endl;
 cout << " Nr pare cu cifra unitatilor " << d << ": " << count;
 return 0;
}
```

48. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 10000000$ ) și determină suma tuturor numerelor naturale echilibrate (un număr este echilibrat dacă are proprietatea că prima cifră a numărului este egală cu ultima cifră a numărului) mai mici sau egale cu  $n$ . Programul afișează pe ecran suma determinată. De exemplu, pentru  $n=17$ , programul va afișa valoarea 56 ( $=1+2+3+\dots+9+11$ ).

| Limbajul Python                                                                                                                                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>n = int(input()) S = 0 for i in range(1, n + 1):     if i &lt; 10: S += i     else:         x = i         while x &gt; 9:             x = x // 10         if x == i % 10:             S += i print(S)</pre> | <pre>#include &lt;iostream&gt; using namespace std; int main() {     int n, S=0, i, x;     cin &gt;&gt; n;     for (i=1; i &lt;= n; i++)         if (i &lt; 10) S = S + i;         else             {x = i;              while (x &gt; 9) x = x / 10;              if (x == i % 10) S = S + i; }     cout &lt;&lt; S;     return 0; }</pre> |

49. Scrieți un program **Python/C++** care să citească două numere naturale  $a$  și  $b$  ( $10 < a < b < 10000$ ) și care să afișeze pe ecran, în ordine crescătoare, separate prin câte un spațiu, toate numerele naturale pare din intervalul  $[a, b]$  care au cifra unităților cu exact două unități mai mare decât cifra zecilor sau mesajul **nu exista**, în cazul în care în intervalul dat nu avem astfel de numere. **De exemplu**, pentru  $a=10$  și  $b=101$ , programul va afișa, pe o linie a ecranului: 24 46 68.

| Limbajul Python                                                                                                                                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a,b = map(int, input().split()) ok=0 if a%2==1: a=a+1 if b%2==1: b=b-1 for c in range(a, b + 1):     if c//10%10+2==c%10 and c%2==0:         print(c, " ")         ok=1 if ok==0:     print("nu exista") </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int a,b,c,ok=0;     cin&gt;&gt;a&gt;&gt;b;     if(a%2==1) a++;     if(b%2==1) b--;     for(c=a;c&lt;=b;c++)         if(c/10%10+2==c%10         &amp;&amp; c%2==0)             {cout&lt;&lt;c&lt;&lt;" ";ok=1;}     if(ok==0) cout&lt;&lt;"nu exista";     return 0; } </pre> |

50. Scrieți un program **Python/C++** care să citească un număr natural  $k$  ( $k < 20$ ) și să afișeze pe ecran primele  $k!$  numere, afișarea se face respectând formatul din exemplu. **De exemplu**, pentru  $k=3$ , programul va afișa:

$0!=1$   
 $1!=1$   
 $2!=2$   
 $3!=6$ .

| Limbajul Python                                                                                       | Limbajul C/C++                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> k=int(input()) p=1 print(0,'!=',p) for c in range(1, k+1):     p=p*c     print(c,'!=',p) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {long long k,c,p=1;     cin&gt;&gt;k;     cout&lt;&lt;0&lt;&lt;"!="&lt;&lt;1&lt;&lt;endl;     for(c=1;c&lt;=k;c++)     {         p=p*c;         cout&lt;&lt;c&lt;&lt;"!="&lt;&lt;p&lt;&lt;endl;     }     return 0; } </pre> |

51. Scrieți un program **Python/C++** care să citească două numere naturale nenule  $n$  și  $k$  ( $n, k < 10000$ ). Programul determină și afișează pe ecran cifra de control a numărului  $n \cdot 10^k$ .

De exemplu, pentru  $n=123$  și  $k=5$ , programul va afișa 6 (deoarece  $1+2+3+0+0+0+0+0=6$  este o cifră, prin urmare cifra de cotrol tot 6), iar pentru  $n=1234$  și  $k=5$ , programul va afișa 1 (deoarece  $1+2+3+4+0+0+0+0+0=10$  nu este o cifră, se calculeaza suma cifrelor noului număr 10, prin urmare cifra de cotrol este 1, pentru că  $1+0=1$ ).

| Limbajul Python                                                                                                                    | Limbajul C/C++                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n,k=map(int,input().split()) s=0 for i in range(1, 5):     s=s+n%10     n=n//10 if s%9==0: print(9) else: print (s%9) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,k,i,s=0;     cin&gt;&gt;n&gt;&gt;k;     for(i=1;i&lt;=4;i++)     { s=s+n%10; n=n/10;}     if(s%9==0) cout&lt;&lt;9;     else cout&lt;&lt;s%9;     return 0; } </pre> |

52. Scrieți un program **Python/C++** care citește un număr natural cu exact 9 cifre. Programul determină și afișează pe ecran toate prefixele numărului, apoi toate prefixele numărului, ca în exemplu

De exemplu, pentru  $n=123456789$ , programul va afișa

123456789

12345678

1234567

123456

12345

1234

123

12

1

1

12

123

1234

12345

123456

1234567

12345678

123456789

| Limbajul Python                                                                                                                                 | Limbajul C/C++                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) c=n p=1000000000 for i in range(1, 10):     print(n)     n=n//10 for i in range(1, 10):     print(c//p)     p=p//10 </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n,k,i,c,p=1000000000;     cin&gt;&gt;n;     c=n;     for(i=1;i&lt;=9;i++)     { cout&lt;&lt;n&lt;&lt;endl; n=n/10; }      for(i=1;i&lt;=9;i++)     { cout&lt;&lt;c/p&lt;&lt;endl; p=p/10; }     return 0; } </pre> |

## V. Șiruri de caractere

1. Se consideră secvența **Python** alăturată. Scrieți ce se va afișa în urma executării secvenței.

```
s = "Examen"
print(s[3])
print(s[2:5])
print(s[:3])
print(s[2:])
print(s[:])
```

2. Se consideră secvența **Python** alăturată. Scrieți ce conținut au variabilele **s** și **t** în urma executării secvenței.

```
a = "Culegere"
b = "Informatica"
s=a+b
t=a+' de ' +b
```

3. Se consideră secvența **Python** alăturată. Scrieți ce se va afișa în urma executării secvenței.

```
a = "Foarte*"
b = "BINE"
c=2*a+b
print(c)
```

4. Știind că **s='VacantaDeVara'**, ce valoare i se va atribui variabilei **t** în urma executării comenzii **Python**: **t= ('x' in s) and ('D' in s) and ('DeV' in s)**

a. xDDeV

b. True

c. False

d. DDev

5. Pentru fiecare secvență **Python** din prima linie a tabelului de mai jos, asociați valoarea afișată în urma executării corespunzătoare din a doua coloană, trasând o linie între cele două valori.

```
a = "Unu doi trei si patru"
x = a.split()
print(x)
```

102

```
b = "Unu, doi, trei și patru"
x = b.split(", ")
print(x)
```

['unu', 'doi', 'trei@patru']

```
a="unu@doi@trei@patru"
x = a.split("@")
print(x)
```

['unu', 'doi', 'trei', 'patru']

```
a="unu@doi@trei@patru"
x = a.split("@",2)
print(x)
```

['Unu', 'doi', 'trei și patru']

```
s = "12 34 56"
x = s.split()
y = int(x[0])
z = int(x[1])
w = int(x[2])
print(y+z+w)
```

['Unu', 'doi', 'trei', 'si', 'patru']

6. Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, doar litere mici ale alfabetului englez. Programul modifică șirul înlocuind fiecare vocală din șir cu litera mare corespunzătoare.

Se consideră vocale literele: 'a', 'e', 'i', 'o', 'u'.

**De exemplu**, dacă se citește șirul **septembrie**, programul va afișa șirul **sEptEmbrIE**

| Limbajul Python                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>s=input() t="" v="aeiou" for i in range(len(s)):     if s[i] in v:         t=t+s[i].upper()     else:         t=t+s[i] s=t print(s)</pre> | <pre>#include &lt;iostream&gt; #include &lt;cstring&gt; using namespace std; int main() {     char s[101];     char voc[]="aeiou";     int i;     cin.get(s,101);     for(i=0; i&lt;strlen(s);i++)         if(strchr(voc, s[i]))             s[i]=toupper(s[i]);     cout&lt;&lt;s;     return 0; }</pre> |

7. Scrieți un program Python/C++ care citește de la tastatură un text format din cel mult 100 de caractere, doar litere ale alfabetului englez și spații. Programul determină numărul vocalelor încadrate de consoane din text. Se consideră vocale literele: 'a', 'e', 'i', 'o', 'u'.

**De exemplu**, pentru textul citit : cartile sunt frumoase programul va afișa: **5**.  
(car, til, sun, rum, moa)

| Limbajul Python                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>s=input() v="aeiou" k = 0 for i in range(1,len(s)-1):     if s[i] in v:         if s[i-1] not in v and s[i-1]!=' ':             if s[i+1] not in v and s[i+1]!=' ':                 k+=1 print (k)</pre>                                                                                                                                                                                        |
| Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                       |
| <pre>#include &lt;iostream&gt; #include &lt;cstring&gt; using namespace std; int main() {     char s[101], voc[]="aeiou";     int i,k=0;     cin.get(s,101);     for(i=1; i&lt;strlen(s)-1;i++)         if( strchr(voc, s[i]) &amp;&amp; !strchr(voc,s[i-1]) &amp;&amp; s[i-1]!=' ' &amp;&amp;             !strchr(voc,s[i+1]) &amp;&amp; s[i+1]!=' ') k++;     cout&lt;&lt;k;     return 0; }</pre> |

8. Scrieți un program **Python/C++** care citește de la tastatură un text format din cel mult 100 de caractere, doar litere ale alfabetului englez, cifre și spații. Programul determină și afișează pe ecran suma tuturor cifrelor din text.

**De exemplu**, pentru textul citit : **ana are 1 mar 2 caise 3 prune si 3 nuci** programul va afișa 9.

| Limbajul Python                                                                                                                   | Limbajul C/C++                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>s=input() cif="0123456789" Sum = 0 for i in range(0,len(s)):     if s[i] in cif:         Sum=Sum+int(s[i]) print (Sum)</pre> | <pre>#include &lt;iostream&gt; #include &lt;cstring&gt; using namespace std; int main() {     char s[101];     char cif[]="0123456789";     int Sum = 0, i;     cin.get(s,101);     for(i=0; i&lt;strlen(s);i++)         if(strchr(cif, s[i]))             Sum=Sum+s[i]-'0';     cout&lt;&lt;Sum;     return 0; }</pre> |

9. Scrieți un program **Python/C++** care citește de la tastatură un text format din cel mult 100 de caractere, doar litere ale alfabetului englez, cifre pare și spații. Programul modifică textul, înlocuind fiecare cifră cu cuvântul corespunzător (0-zero, 2-doi, 4-patru, 6-sase, 8-opt). Programul afișează pe ecran textul modificat.

**De exemplu**, pentru textul citit: **ana are 2 mere 4 caise 6 prune și 8 nuci** programul va afișa:

**ana are doi mere patru caise sase prune și opt nuci.**

| Limbajul Python                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>s=input() c2="doi" c4="patru" c6="sase" c8="opt" cif="2468" t="" for i in range(0,len(s)):     if s[i] in cif:         if s[i]=='2': t+=c2         else:             if s[i]=='4': t+=c4             else:                 if s[i]=='6': t+=c6                 else: t+=c8     else: t+=s[i] s=t print (s)</pre> |
| Limbajul C/C++                                                                                                                                                                                                                                                                                                        |
| <pre>#include &lt;iostream&gt; #include &lt;cstring&gt;</pre>                                                                                                                                                                                                                                                         |

```

using namespace std;
int main()
{
 char s[101], t[101]="";
 char cif[]="2468";
 int i,k;
 cin.get(s,101);
 for(i=0; i<strlen(s);i++)
 if(strchr(cif, s[i]))
 switch (s[i])
 {
 case '2': strcat(t,"doi");break;
 case '4': strcat(t,"patru");break;
 case '6': strcat(t,"sase");break;
 case '8': strcat(t,"opt");break;
 }
 else { k=strlen(t);
 t[k++]=s[i];
 t[k]='\0';
 }
 strcpy(s,t);
 cout<<s;
 return 0;
}

```

10. Scrieți un program **Python/C++** care citește de la tastatură un text format din cuvinte separate prin spații. Textul conține cel mult 100 de caractere, iar cuvintele sunt formate doar din litere mici ale alfabetului englez. Programul modifică textul, ștergând ultima literă a fiecărui cuvântul care se termină cu o consoană. Programul afișează pe ecran textul modificat.

**De exemplu**, pentru textul citit: **ana are un caiet si un pix**

programul va afișa: **ana are u caie si u pi**

| Limbajul Python                                                                                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> s=input() t="" svoc=" aeiou" for i in range(0,len(s)-1):     if s[i] not in svoc:         if s[i+1]!=' ': t=t+s[i]     else: t=t+s[i] s=t print (s) </pre> | <pre> #include &lt;iostream&gt; #include &lt;cstring&gt; using namespace std; int main() {     char s[101], t[101];     char svoc[]=" aeiou";     int i,k=0;     cin.get(s,101);     for(i=0; i&lt;strlen(s)-1;i++)         if(strchr(svoc, s[i])==NULL)             {if( s[i+1]!=' ')t[k++]=s[i];}         else t[k++]=s[i];     t[k]='\0';strcpy(s,t);     cout&lt;&lt;s;     return 0; } </pre> |

11. Scrieți un program **Python/C++** care citește de la tastatură două cuvinte, formate din cel mult 50 caractere doar litere ale alfabetului englez, și afișează pe ecran mesajul **DA** dacă acestea sunt **anagrame**. În caz contrar, va afișa mesajul **NU**.

**De exemplu**, dacă se citesc cuvintele **abcbfde** și **bdfcabe** programul va afișa mesajul **DA**, iar pentru **abca** și **abcd** va afișa mesajul **NU**.

| Limbajul Python                                                                                                                                | Limbajul C/C++                                                                                                                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> a=input() b=input() a_sort = ''.join(sorted(a)) b_sort = ''.join(sorted(b)) if a_sort==b_sort:     print("DA") else : print("NU") </pre> | <pre> #include &lt;iostream&gt; #include &lt;algorithm&gt; #include &lt;cstring&gt; using namespace std; int main() {     char a[51], b[51];     cin&gt;&gt;a&gt;&gt;b;     sort(a,a+strlen(a));     sort(b,b+strlen(b));     if(strcmp(a,b)==0) cout&lt;&lt;"DA";     else cout&lt;&lt;"NU";     return 0; } </pre> |

12. Scrieți un program **Python/C++** care citește de la tastatură un șir de cel mult 100 de caractere, litere mici ale alfabetului englez și care afișează pe ecran, pe același rând, separate prin câte un spațiu, vocalele distincte apărute în șir, în ordine alfabetică.

**Exemplu:** pentru șirul `invat python` se afișează `a i o`

| Limbajul Python                                                                                                                                                                                                                         | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> sir = input() vocale = {'a', 'e', 'i', 'o', 'u'} vocale_gasite = set() for caracter in sir:     if caracter in vocale:         vocale_gasite.add(caracter) for vocala in sorted(vocale_gasite):     print(vocala, end=' ') </pre> | <pre> #include &lt;iostream&gt; #include &lt;cstring&gt; using namespace std; int main() {     char text[101];     char vocale[]="aeiou";     bool aparitii[5]={false};     cin.getline(text,101);     for(int i=0; i&lt;5;i++)         if(strchr(text, vocale[i]))             aparitii[i]=true;     for (int i=0;i&lt;5;i++)         if (aparitii[i])             cout&lt;&lt;vocale[i]&lt;&lt;' ';     return 0; } </pre> |

13. Scrieți un program **Python/C++** care citește de la tastatură un șir de cel mult 100 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran pe rânduri diferite, vocalele apărute în șir, în ordine alfabetică, pentru fiecare vocală afișându-se numărul de apariții, ca în exemplu.

**Exemplu:** pentru textul `invata python ca sa fii destept` se afișează:

```
a: 3
e: 2
i: 2
o: 1
u: 1
```

### Limbajul Python

```
sir = input()
voc = ['a', 'e', 'i', 'o', 'u']
frecventa = {vocala: 0 for vocala in voc}
for caracter in sir:
 if caracter in frecventa:
 frecventa[caracter] += 1
for vocala in voc:
 if frecventa[vocala] > 0:
 print(f"{vocala}: {frecventa[vocala]}")
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char text[101];
 int freqv[256]={0};
 cin.getline(text,101);
 for (int i=0;text[i];i++)
 {
 char ch = text[i];
 if (ch>='a' && ch<='z') freqv[(int)ch]++;
 }
 char voc[]="aeiou";
 for (int i=0;i<5;i++)
 {
 char v = voc[i];
 if (freqv[(int)v]>0)
 {cout<<v<<" ";
 cout<<freqv[(int)v]<<"\n";}
 }
 return 0;
}
```

14. Scrieți un program **Python/C++** care citește de la tastatură un șir de cel mult 100 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran, textul obținut în urma înlocuirii fiecărei consoane cu vocala imediat dinaintea ei în alfabet și a fiecărei vocale cu litera imediat următoare în alfabet.

**Exemplu:** pentru textul `eu invata python` se afișează `fv jiubo ouoepe`

## Limbajul Python

```
vocale = {'a', 'e', 'i', 'o', 'u'}

Funcție pentru înlocuire
def transforma(caracter):
 if caracter in vocale:
 return chr(ord(caracter) + 1)
 elif 'a' <= caracter <= 'z':
 for i in range(ord(caracter) - 1, ord('a') - 1, -1):
 if chr(i) in vocale:
 return chr(i)
 return caracter
 else:
 return caracter

sir = input()
sir_nou = ''.join(transforma(c) for c in sir[:100])
print(sir_nou)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
bool Vocala(char ch)
{
 if(strchr("aeiou",ch)) return 1;
 return 0;
}
char vocalaDinainte(char ch)
{
 for (char c=ch-1;c>='a';c--)
 if (Vocala(c))
 return c;
 return ch;
}
int main()
{
 char text[101];
 cin.getline(text, 101);
 for (int i=0;text[i];++i)
 {
 char ch=text[i];
 if (ch==' ')
 cout<<" ";
 else if(Vocala(ch))
 {
 if(ch=='u') cout<<'v';
 else
 cout<<char(ch+1);
 }
 else
 cout<<vocalaDinainte(ch);
 }
 cout<<'\n';
 return 0;
}
```

15. Scrieți un program **Python/C++** care citește de la tastatură un șir de cel mult 100 de caractere, litere mari și mici ale alfabetului englez și spații și care afișează pe ecran textul obținut în urma înlocuirii fiecărei litere mari cu litera mică corespunzătoare ei și fiecare literă mică cu litera mare corespunzătoare ei.

**Exemplu:** dacă șirul este `InvAT pYthON` se va afișa `iNVat PyTHon`

#### Limbajul Python

```
sir = input()
sir_transformat = ''
for caracter in sir:
 if caracter.islower():
 sir_transformat += caracter.upper()
 elif caracter.isupper():
 sir_transformat += caracter.lower()
 else:
 sir_transformat += caracter
print(sir_transformat)
```

#### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
char sir[101];
int main()
{
 cin.getline(sir,101);
 for (int i=0; sir[i]; ++i)
 if (isupper(sir[i]))
 sir[i]=tolower(sir[i]);
 else if (islower(sir[i]))
 sir[i]=toupper(sir[i]);
 cout<< sir << '\n';
 return 0;
}
```

16. Scrieți un program **Python/C++** care citește de la tastatură un text de cel mult 200 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran cuvintele, separate printr-un spațiu, care încep și se termină cu aceeași literă.

**Exemplu:** dacă textul citit este

ana si dana sunt eleve de o nota buna

se afișează: `ana eleve o`

#### Limbajul Python

```
text = input()
cuvinte = text.split()
rezultat = [cuvant for cuvant in cuvinte if cuvant[0] == cuvant[-1]]
print(" ".join(rezultat))
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
bool litere(const char* cuv)
{ int lg=strlen(cuv);
 if (lg==0) return false;
 return cuv[0]==cuv[lg-1];
}
int main()
{ char text[201];
 cin.getline(text, 201);
 char* cuv=strtok(text, " ");
 bool prima=true;

 while (cuv!=NULL)
 { if (litere(cuv))
 { if (!prima)cout<<" ";
 cout<<cuv;
 prima=false;
 }
 cuv=strtok(NULL, " ");
 }
 cout<<'\n';
 return 0;
}
```

17. Scrieți un program **Python/C++** care citește de la tastatură un text de cel mult 200 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran cuvintele, separate printr-un spațiu, care conțin o singură vocală, indiferent de numărul de apariții ale acesteia.

De exemplu, pentru textul: dana si costel au folos si noroc la plaja  
se afișează `dana si folos si noroc la plaja`

## Limbajul Python

```
vocale = {'a', 'e', 'i', 'o', 'u'}
text = input()
cuvinte = text.split()
rezultat = []
for cuvant in cuvinte:
 vocale_gasite = set([litera for litera in cuvant if litera in vocale])
 if len(vocale_gasite) == 1:
 rezultat.append(cuvant)
print(" ".join(rezultat))
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
#include <set>
using namespace std;
bool esteVocala(char ch)
{
 if(strchr("aeiou",ch)) return 1;
 return 0;
}
```

```

bool oSinguraVocala(const char* cuv)
{
 set<char> vocale;
 for (int i=0;cuv[i];i++)
 if (esteVocala(cuv[i]))
 { vocale.insert(cuv[i]);
 if (vocale.size()>1) return false;
 }
 return !vocale.empty();
}

int main()
{
 char text[201];
 cin.getline(text,201);
 char* cuv=strtok(text," ");
 bool prima=true;
 while (cuv!=NULL)
 { if (oSinguraVocala(cuv))
 { if (!prima)cout<<" "; cout<<cuv;
 prima=false; }
 cuv=strtok(NULL," ");
 }
 cout<<'\n';
 return 0; }

```

18. Scrieți un program **Python/C++** care citește de la tastatură un text de cel mult 200 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran textul obținut după înlocuirea fiecărui cuvânt cu număr par de litere cu un cuvânt obținut din acesta format doar din primele două litere ale acestuia. Cuvintele sunt separate printr-un singur spațiu.

**De exemplu**, pentru textul invat python si rezolv multe probleme se afișează: invat py si re multe pr

#### Limbajul Python

```

text = input()
cuvinte = text.split()
rezultat = []
for cuvant in cuvinte:
 if len(cuvant) % 2 == 0:
 rezultat.append(cuvant[:2])
 else:
 rezultat.append(cuvant)
print(" ".join(rezultat))

```

#### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
void prelCuv(char* cuv)
{
 int len=strlen(cuv);
 if (len %2==0 && len>=2)
 cuv[2]='\0';
}

```

```

int main()
{
 char text[201];
 cin.getline(text,201);
 char* cuv=strtok(text," ");
 bool prima=true;
 while (cuv!=NULL)
 {
 prelCuv(cuv);
 if (!prima) cout<<" ";
 cout<<cuv;
 prima=false;
 cuv=strtok(NULL, " ");
 }
 cout<<'\n';
 return 0;
}

```

19. Scrieți un program **Python/C++** care citește de la tastatură un text de cel mult 200 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran textul obținut după eliminarea din fiecare cuvânt cu cel puțin trei litere a literelor aflate pe poziție pară (a doua literă, a patra literă, ș.a.m.d). Cuvintele sunt separate printr-un singur spațiu.

**De exemplu,** pentru textul un copil care invata multe lucruri este destept

Se afișează: un cpl cr ivt mle lcui et dset

#### Limbajul Python

```

text = input()
cuvinte = text.split()
rezultat = []
for cuv in cuvinte:
 if len(cuv) >= 3:
 cuv_nou=''
 for i in range(len(cuv)):
 if i%2 == 0:
 cuv_nou+=cuv[i]
 rezultat.append(cuv_nou)
 else:
 rezultat.append(cuv)
print(" ".join(rezultat))

```

#### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
void eliminaPozitiiPare(char* cuv)
{
 int len=strlen(cuv), j=0;
 if (len<3) return;
 for (int i=0;i<len;++i)
 if (i%2==0) cuv[j++]=cuv[i];
 cuv[j]='\0';
}

```

```

int main()
{
 char text[201];
 cin.getline(text, 201);
 char* cuv= strtok(text, " ");
 bool prima=true;
 while (cuv!=NULL)
 {
 eliminaPozitiiPare(cuv);
 if(!prima) cout<<" "; cout<<cuv;
 prima=false;
 cuv= strtok(NULL, " ");
 }
 cout<<'\n';
 return 0;
}

```

20. Scrieți un program **Python/C++** care citește de la tastatură un text de cel mult 200 de caractere, litere mici ale alfabetului englez și spații și care afișează pe ecran textul obținut după răsturnarea cuvintelor care conțin litera a. Cuvintele sunt separate printr-un singur spațiu.

**De exemplu,** pentru textul `limbajul python este interesant`

Se afișează `lujabmil python este tnaseretni`

#### Limbajul Python

```

text = input()
cuvinte = text.split()
rezultat = []
for cuv in cuvinte:
 if 'a' in cuv:
 rezultat.append(cuv[::-1]) # răstoarnă cuvântul
 else:
 rezultat.append(cuv)
print(" ".join(rezultat))

```

#### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;

bool contineLiteraA(const char* cuv)
{
 for (int i=0; cuv[i]; i++)
 if (cuv[i]=='a') return true;
 return false;
}

void rastornaCuvant(char* cuv)
{
 int st=0, dr=strlen(cuv)-1;
 while (st < dr)
 {
 char temp=cuv[st]; cuv[st]=cuv[dr]; cuv[dr]=temp;
 st++; dr--;
 }
}

int main()
{
 char text[201];
 cin.getline(text, 201);

```

```

char* cuv=strtok(text," ");
bool prima=true;
while (cuv!=NULL)
{ if (contineLiteraA(cuv)) rastoarnaCuvant(cuv);
 if (!prima)cout<<" ";
 cout<<cuv;
 prima=false; cuv=strtok(NULL," ");
}
cout <<'\n';
return 0;
}

```

21. Scrieți un program **Python/C++** care citește de la tastatură un text de cel mult 100 de caractere, litere mici ale alfabetului englez, spații și cifre și care afișează pe ecran, pe linii separate, fiecare literă în ordine alfabetică și frecvența ei de apariție și apoi fiecare cifră și frecvența ei de apariție, ca în exemplu. Se vor afișa doar literele și cifrele care apar. **De exemplu**, pentru textul:

un elev bun ia 10 altul ia 2 de 7 si altul 4 de 2  
se afișează:

```

a 4
b 1
d 2
e 4
i 3
l 5
n 2
s 1
t 2
u 4
v 1
0 1
1 1
2 2
4 1
7 1

```

### Limbajul Python

```

from collections import Counter
text = input()
freq = Counter(text)
litere = {ch: cnt for ch, cnt in freq.items() if ch.isalpha()}
cifre = {ch: cnt for ch, cnt in freq.items() if ch.isdigit()}
for litera in sorted(litere):
 print(f"{litera} {litere[litera]}")
for cifra in sorted(cifre):
 print(f"{cifra} {cifre[cifra]}")

```

### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{ char text[101];
 int frec_litere[26]={0}, frec_cifre[10]={0};
 cin.getline(text, 101);
 for (int i=0;text[i]!='\0';++i)
 { if (text[i]>='a' && text[i]<='z') frec_litere[text[i]-'a']++;
 else if (text[i]>='0' && text[i]<='9')
 frec_cifre[text[i]-'0']++; }
}

```

```

for (int i=0;i<26;++i)
 if (frec_litere[i]>0)cout<<char('a'+i)<<" "<<frec_litere[i]<<'\n';
for (int i=0;i<10;++i)
 if (frec_cifre[i] > 0) cout<<i<<" "<< frec_cifre[i]<<'\n';
return 0;
}

```

22. Scrieți un program **Python/C++** care citește de la tastatură un text format din litere mari și mici ale alfabetului englez, cifre, spații și semne de punctuație (cel mult 256 de caractere). Programul, considerând literele mari ca fiind identice cu cele mici, va determina consoana care apare de cele mai multe ori în text. Dacă există mai multe consoane cu același număr maxim de apariții, programul va afișa pe ecran prima consoană în ordine alfabetică.

**Exemplu:** Pentru textul introdus: **"Programarea în C++ este captivantă!"** programul va afișa: **c**

### Limbaajul Python

```

text = input("Introduceți textul: ")
text = text.lower()
frecv = [0] * 26
for ch in text:
 if ch.isalpha() and ch not in 'aeiou':
 index = ord(ch) - ord('a')
 frecv[index] += 1
maxim = max(frecv)
Căutăm prima consoană în ordine alfabetică cu frecvență maximă
for i in range(26):
 if frecv[i] == maxim:
 litera = chr(i + ord('a'))
 break
print("Consoana cu cele mai multe apariții:", litera)

```

### Limbaajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
int main() {
 char text[256];
 cin.getline(text, 256);
 int frecv[26] = {0};
 for (int i = 0; text[i]; i++) {
 char ch = tolower(text[i]);
 if (isalpha(ch) && !(ch == 'a' || ch == 'e' || ch == 'i'
 || ch == 'o' || ch == 'u')) frecv[ch - 'a']++;
 }
 int maxim = 0;
 for (int i = 0; i < 26; i++) {
 if (frecv[i] > maxim) maxim = frecv[i];
 }
 for (int i = 0; i < 26; i++) {
 if (frecv[i] == maxim) {
 cout << "Consoana cu cele mai multe apariții: ";
 cout << char('a' + i) << endl; break; }
 }
 return 0;
}

```

23. Scrieți un program **Python/C++** care citește un cuvânt (format din maxim 256 de caractere) și verifică dacă poate deveni palindrom prin eliminarea unui singur caracter, situație în care programul va afișa pe ecran mesajul **DA**, altfel **NU**. Cuvântul citit este format din cel mult 256 de caractere, doar litere mici ale alfabetului englez.

**Exemplu:** dacă se citește cuvântul **radar**, programul va afișa mesajul **DA**, iar pentru cuvântul **tacobell**, va afișa **NU**

### Limbajul Python

```
def este_palindrom(s):
 return s == s[::-1]

def poate_deveni_palindrom(cuvant):
 for i in range(len(cuvant)):
 nou = cuvant[:i] + cuvant[i+1:]
 if este_palindrom(nou):
 return True
 return False

cuvant = input("Introduceți cuvântul: ").lower()
print("DA" if poate_deveni_palindrom(cuvant) else "NU")
```

### Limbajul C++

```
#include <iostream>
#include <cstring>
using namespace std;

bool este_palindrom_fara(char s[], int n, int skip) {
 int i = 0, j = n - 1;
 while (i < j) {
 if (i == skip) i++;
 // Sărim peste caracterul de eliminat
 if (j == skip) j--;
 if (i < j && s[i] != s[j]) return false;
 i++; j--;
 }
 return true;
}

int main() {
 char cuv[256];
 cin >> cuv;
 int n = strlen(cuv);

 for (int i = 0; i < n; i++) {
 // Încercăm să eliminăm fiecare caracter pe rând
 if (este_palindrom_fara(cuv, n, i)) {
 cout << "DA\n";
 return 0;
 }
 }
 cout << "NU\n";
 return 0; }
```

24. Scrieți un program **Python/C++** care citește o propoziție, apoi modifică propoziția prin înlocuirea tuturor spațiilor cu caracterul `'_'`, iar literele mici sunt transformate în majuscule. Programul va afișa pe ecran propoziția modificată. Propoziția citită este formată din cel mult 256 de caractere, doar litere mici ale alfabetului englez și spații. Exemplu: pentru propoziția "salut lume" , programul va afișa **SALUT\_LUME**

#### Limbajul Python

```
text = input()
rezultat = ""
for ch in text:
 if ch == ' ':
 rezultat += '_'
 else:
 rezultat += ch.upper()

print(rezultat)
```

#### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
 char text[256];
 cin.getline(text, 256);
 for (int i = 0; text[i] != '\0'; i++) {
 if (text[i] == ' ') cout << "_";
 else cout << (char)toupper(text[i]);
 }
 cout << endl;
 return 0;
}
```

25. Scrieți un program **Python/C++** care citește o propoziție reprezentând parola magică rostită de un spiriduș păzitor. Pentru a o descifra, programul trebuie să construiască un nou șir format doar din prima literă a fiecărui cuvânt din propoziție, transformate în majuscule și să afișeze pe ecran noul șir. Propoziția citită este formată din cel mult 256 de caractere, doar litere mici ale alfabetului englez și spații. Exemplu: pentru textul "pădurea de argint" se va afișa **PDA**

#### Limbajul Python

```
text = input()
text = text.strip()
Construim rezultatul
parola = ""
cuvinte = text.split()
for cuv in cuvinte:
 if cuv: # verificăm dacă cuvântul nu e gol
 parola += cuv[0].upper()

print(parola)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;

int main() {
 char text[256];
 cin.getline(text, 256);
 bool inceput_cuvant = true;
 for (int i = 0; text[i]; i++) {
 if (text[i] == ' ') {
 inceput_cuvant = true;
 } else if (inceput_cuvant) {
 cout << (char)toupper(text[i]);
 inceput_cuvant = false;
 }
 }
 cout << endl;
 return 0;
}
```

26. Scrieți un program **Python/C++** care citește un cuvânt rostit de un erou în fața oglinzii de cristal. Oglinda creează un cuvânt nou format din cuvântul inițial urmat de forma sa inversă. Programul construiește noul cuvânt și îl afișează pe ecran. Cuvântul citit este format din cel mult 100 de caractere, doar litere mici ale alfabetului englez.

**Exemplu:** pentru cuvântul citit **foc**, programul va afișa cuvântul **foccof**

## Limbajul Python

```
cuvant = input("Introduceți cuvântul: ")
Inversare manuală
invers = ""
for i in range(len(cuvant) - 1, -1, -1):
 invers += cuvant[i]
Concatenăm cuvântul original cu inversul
rezultat = cuvant + invers
print(rezultat)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
 char cuv[101], rezultat[201];
 cin >> cuv;
 int n = strlen(cuv);
 // Copiem cuvântul inițial
 strcpy(rezultat, cuv);
 // Adăugăm forma inversă
 for (int i = n - 1; i >= 0; i--)
 rezultat[n + (n - 1 - i)] = cuv[i];
}
```

```

rezultat[2 * n] = '\0'; // terminator de șir
cout << rezultat << endl;
return 0;
}

```

27. Scrieți un program **Python/C++** care citește o propoziție din jurnalul unui magician și calculează și afișează pe ecran câte dintre cuvinte încep cu o consoană și câte cu o vocală. Propoziția citită este formată din cel mult 256 de caractere, doar litere mici ale alfabetului englez și spații.

**Exemplu:** dacă se citește propoziția: **aceasta e ziua cand zalmoxe a apărut**, programul va afișa: **Vocale: 3, Consoane: 3**

### Limbajul Python

```

text = input()
vocale = "aeiou"
count_vocale = 0
count_consoane = 0
cuvinte = text.split()
for cuv in cuvinte:
 if cuv:
 prima = cuv[0].lower()
 if prima.isalpha():
 if prima in vocale: count_vocale += 1
 else: count_consoane += 1
print(f"Vocale: {count_vocale}, Consoane: {count_consoane}")

```

### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;

bool este_vocala(char ch) {
 ch = tolower(ch);
 return ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u';
}

int main() {
 char text[256]; int vocale = 0, consoane = 0;
 cin.getline(text, 256);
 bool inceput_cuvant = true;
 for (int i = 0; text[i]; i++) {
 if (isalpha(text[i])) {
 if (inceput_cuvant) {
 if (este_vocala(text[i])) vocale++;
 else consoane++;
 inceput_cuvant = false;
 } else inceput_cuvant = true;
 }
 }
 cout << "Vocale: " << vocale << ", Consoane: " << consoane << endl;
 return 0;
}

```

28. Scrieți un program **Python/C++** care citește un text reprezentând o comandă rostită de pilotul unui OZN și verifică dacă toate cuvintele încep cu aceeași literă. În caz afirmativ, programul va afișa mesajul **ACTIVAT**, altfel va afișa mesajul **EROARE**. Textul citit este format din cel mult 256 de caractere, doar litere mici ale alfabetului englez și spații.

**Exemplu:** dacă se citește textul: **zorzon zare zibri** atunci programul va afișa mesajul: **ACTIVAT**

#### Limbajul Python

```
text = input()
cuvinte = text.split()
initiala = cuvinte[0][0].lower()
activat = True
for cuvant in cuvinte:
 if cuvant[0].lower() != initiala:
 activat = False
 break
if activat:
 print("ACTIVAT")
else:
 print("EROARE")
```

#### Limbajul C/C++

```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

int main() {
 char text[256];
 cin.getline(text, 256);
 char initiala = '\0';
 bool ok = true;
 for (int i = 0; text[i]; i++) {
 if (i == 0 || (text[i - 1] == ' ' && isalpha(text[i]))) {
 char lit = tolower(text[i]);
 if (initiala == '\0') initiala = lit;
 else if (lit != initiala) {
 ok = false; break;
 }
 }
 }
 cout << (ok ? "ACTIVAT" : "EROARE");
 return 0;
}
```

29. Scrieți un program în **Python/C++** care citește de la tastatură un cuvânt format din litere mari și mici ale alfabetului englez. Programul va trata literele mari și mici ca fiind identice, va determina și va afișa, pe ecran, numărul de consoane din cuvânt.

Se consideră consoane toate literele mici din alfabetul englez care nu sunt vocale (a, e, i, o, u).

De exemplu, pentru bacalaureat, se va afișa 5.

#### Limbajul Python

```
cuvant = input("Introduceți un cuvânt: ")
vocale = "aeiou"
count = 0
for litera in cuvant:
 if litera.isalpha() and litera not in vocale:
 count += 1
print("Numărul de consoane este:", count)
```

#### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
 char a[100];
 int i, count = 0;
 cout<<"Introduceți un cuvânt: ";
 cin >> a;
 for (i = 0; i < strlen(a); i++) {
 if (isalpha(a[i]) && strchr("aeiou", a[i]) == 0)
 count++;
 }
 cout << "Nr. de consoane este: " << count;
 return 0;
}
```

30. Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, litere mici, litere mari, spații și semne de punctuație ale alfabetului englez. Să se modifice și să se afișeze textul astfel încât toate literele mari vor fi transformate în litere mici, toate semnele de punctuație vor fi înlocuite cu caracterul \* și toate celelalte caractere (litere mici, spații, cifre etc.) rămân nemodificate.

De exemplu, pentru: Salut, Lume! 2024 este aici.

se va afișa: salut\* lume\* 2024 este aici\*

#### Limbajul Python

```
s=input("Introdu propoziția: ")
rez = ""
i=0
```

#### Limbajul C/C++

```
#include <iostream>
#include <string>
#include <cctype>
```

```

while i < len(propozitie):
 c = propozitie[i]
 if c.isupper():
 rez += c.lower()
 elif c in ".,!?:":
 rez += '*'
 else:
 rez += c
 i+=1
print("Rezultatul este:", rez)

```

```

using namespace std;
int main() {
 char s[100], rez[] = "",c;
 cout << "Intro propoziția: ";
 getline(cin, s);
 i=0;
 while(i<strfeng(s)){
 c=s[i];
 if (isupper(c))rez += tolower(c);
 else
 if(c=='.'||c==','||c=='!'||c=='?')
 rez += '*';
 else rez += c;
 i++;
 }
 cout << "Rezultatul este: " << rez;
 return 0;
}

```

31. Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, doar litere mici ale alfabetului englez și spații. Programul determină numărul literelor mici din șir situate între două spații sau între un spațiu și marginea șirului, apoi afișează pe ecran acest număr.

**De exemplu**, pentru textul "a b cDe f g ", se va afișa: 3 ( literele a,b și g se afla între spații)

### Limbaajul Python

```

text =input("Introdu textul: ")
count = 0
length = len(text)
for i in range(length):
 if text[i].islower():
 if (i==0 or text[i-1]==' ') and (i==length - 1 or text[i+1]==' '):
 count += 1
print("Număr de litere mici între spații sau margini:", count)

```

### Limbaajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
int main() {
 char text[101];
 boole i_spatiu, d_spatiu;
 cout << "Introduceți textul: ";
 cin.getline(text, 101);
 int count = 0;
 int len = strlen(text);

```

```

for (int i = 0; i < len; i++) {
 if (islower(text[i])) {
 i_spatiu=(i==0)|| (text[i - 1]==' ');
 d_spatiu=(i==len - 1)|| (text[i + 1] == ' ');
 if (i_spatiu && d_spatiu)
 count++;
 }
}
cout << "Număr de litere mici între spații sau margini: " << count;
return 0;
}

```

32. Un erou dintr-o legendă citește un cuvânt scris pe o stâncă magică. Stânca creează un efect special: afișează un **șir nou** format astfel: **prima literă a cuvântului**, apoi **cuvântul inversat**, urmat de **ultima literă a cuvântului original**.

Scrieți un program **Python/C++** care citește cuvântul și afișează „reflexia dublă” generată de stâncă.

**De exemplu**, pentru cuvântul **dragon**, se va afișa: **dnogardn**

#### Limbajul Python

```

def inverseaza(cuvant):
 return cuvant[::-1]

cuv = input("Introduceți cuvântul: ")
if len(cuv) >= 2:
 inv = inverseaza(cuv)
 rezultat = cuv[0] + inv + cuv[-1]
 print("Reflexia dublă este:", rezultat)
else:
 rezultat = cuv * 3
 print("Reflexia simplă este:", rezultat)

```

#### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
void inverseaza(char original[], char invers[]) {
 int n = strlen(original);
 for (int i = 0; i < n; i++) invers[i] = original[n - 1 - i];
 invers[n] = '\0';
}

int main() {
 char cuvant[101];
 char invers[101];
 char rezultat[205];

```

```

cout << "Introduceti cuvantul: ";
cin >> cuvant;
int lungime = strlen(cuvant);
if (lungime >= 2) {
 inverseaza(cuvant, invers);
 rezultat[0] = cuvant[0]; // prima literă

 rezultat[1] = '\0';
 strcat(rezultat, invers); // + inversul
 int lenRez = strlen(rezultat);
 rezultat[lenRez] = cuvant[lungime - 1]; // + ultima literă
 rezultat[lenRez + 1] = '\0';
 cout << "Reflexia dubla este: " << rezultat << endl;
} else {
 rezultat[0] = cuvant[0];
 rezultat[1] = cuvant[0];
 rezultat[2] = cuvant[0];
 rezultat[3] = '\0';

 cout << "Reflexia simpla este: " << rezultat << endl;
}
return 0;
}

```

33. Un agent spațial primește o secvență de cuvinte criptate. El trebuie să verifice dacă **toate cuvintele din mesaj au aceeași lungime**. Dacă condiția se îndeplinește atunci se va afișa mesajul **COD VALID**, altfel se va afișa mesajul **COD INVALID**

Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, doar litere mici ale alfabetului englez și spații și afișează mesajul corespunzător.

**De exemplu**, pentru textul **alfa beta gama**, se va afișa pe ecran : **COD VALID**

pentru textul **start stop stare stea**, se va afișa pe ecran : **COD INVALID**

#### Limbajul Python

```

text = input("Introduceți mesajul criptat: ").strip()
cuvinte = text.split()

lungime = len(cuvinte[0])
valid = all(len(cuvant) == lungime for cuvant in cuvinte)

print("COD VALID" if valid else "COD INVALID")

```

#### Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;

```

```

int main() {
 char sir[256];
 cout << "Introduceti mesajul criptat: ";
 cin.getline(sir, 256);
 int i = 0, lungimeCurenta = 0, lungimeInitiala = -1;
 bool valid = true;
 while (sir[i]) {
 while (sir[i] == ' ') i++;
 if (sir[i] == '\0') break;
 lungimeCurenta = 0;
 while (sir[i] != '\0' && sir[i] != ' ') {
 lungimeCurenta++; i++;
 }
 if (lungimeInitiala == -1) {
 lungimeInitiala = lungimeCurenta;
 } else {
 if (lungimeCurenta != lungimeInitiala) {
 valid = false; break;
 }
 }
 }
 cout << (valid ? "COD VALID" : "COD INVALID");
 return 0;
}

```

34. Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, doar litere mici ale alfabetului englez și spații. Să se afișeze aceeași propoziție, în care literele din fiecare cuvânt sunt inversate, dar ordinea cuvintelor rămâne neschimbată.

**De exemplu**, pentru textul **mama are mere**, se va afișa pe ecran : **amam era erem**

| Limbajul Python                                                                                                                                                                                              | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> prop=input("Introduceți propoziția: ") cuv = prop.split() cuv_inv = [] for cuvant in cuv:     inv=cuvant[::-1] cuv_inv.append(inversat) rez = ' '.join(cuv_inv)  print("Prop finala este:",rez) </pre> | <pre> #include &lt;iostream&gt; #include &lt;cstring&gt; #include &lt;algorithm&gt; using namespace std; int main() {     char linie[256];     cout &lt;&lt; "Introduceți propoziția: ";     cin.getline(linie, 256);     char* cuvant = strtok(linie, " ");     while (cuvant != NULL) {         int len = strlen(cuvant);         reverse(cuvant, cuvant + len);         cout &lt;&lt; cuvant &lt;&lt; " ";         cuvant = strtok(NULL, " ");     }     return 0; } </pre> |

35. Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, doar litere mici ale alfabetului englez, separate prin spații care **să afișeze cuvintele ordonate alfabetic descrescător**, dar **fiecare scris cu litere mari**.

De exemplu, pentru textul ana iubeste muzica si dansul,  
se va afișa: SI MUZICA IUBESTE DANSUL ANA

### Limbajul Python

```
propozitie = input().strip()
cuvinte = propozitie.split()
cuvinte.sort(reverse=True)
cuvinte_mari = [c.upper() for c in cuvinte]
print(" ".join(cuvinte_mari))
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

int main() {
 char linie[101], *p, cuvinte[101][21];
 int nr = 0;
 cin.getline(linie, 101);
 p = strtok(linie, " ");
 while (p != NULL) {
 strcpy(cuvinte[nr], p); nr++; p = strtok(NULL, " ");
 }
 for (int i = 0; i < nr - 1; i++) {
 for (int j = i + 1; j < nr; j++) {
 if (strcmp(cuvinte[i], cuvinte[j]) < 0) {
 char aux[50];
 strcpy(aux, cuvinte[i]);
 strcpy(cuvinte[i], cuvinte[j]);
 strcpy(cuvinte[j], aux);
 }
 }
 }
 for (int i = 0; i < nr; i++) {
 for (int j = 0; j < strlen(cuvinte[i]); j++)
 cuvinte[i][j] = toupper(cuvinte[i][j]);
 cout << cuvinte[i] << " "; }
 return 0;
}
```

36. Se dau două șiruri de caractere formate din litere mici ale alfabetului englez. Scrieți un program **Python/C++** care citește de la tastatură două șiruri de caractere e formate din cel mult 100 de caractere, doar litere mici ale alfabetului englez și spații, și

afișează toate caracterele distincte din al doilea șir care apar și în primul șir, în ordinea apariției lor în al doilea șir.

**De exemplu**, pentru textul `sir1 = "echipament"` și `sir2 = "teatru"`

se va afișa: `t e a u`.

### Limbajul Python

```
sir1=input("Introduceți primul șir: ").strip()
sir2=input("Introduceți al doilea șir:").strip()
afisate = set()
for lit in sir2:
 if lit in sir1 and lit not in afisate:
 print(lit, end=" ")
 afisate.add(lit)
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
 char sir1[101], sir2[101], c;
 bool dejaAfis[256] = {false};
 cout << "Introdu primul șir: ";
 cin.getline(sir1, 101);
 cout << "Introdu al doilea șir: ";
 cin.getline(sir2, 101);
 cout << "Caractere comune unice în ordinea din al doilea șir: ";
 for (int i=0;i<strlen(sir2);i++){
 c = sir2[i];
 if(strchr(sir1,c)
 if(!dejaAfis[(unsigned char)c]){
 cout << c << " "; dejaAfis[(unsigned char)c]=true; }
 }
 return 0;
}
```

37. Un arheolog a descoperit o **inscripție străveche** formată din cuvinte scrise cu litere mici și separate prin spațiu. Unele cuvinte par să fie „**reflexii perfecte**” – adică se citesc la fel de la stânga la dreapta și de la dreapta la stânga.

Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 100 de caractere, doar litere mici și mari ale alfabetului englez și spații, care identifică toate aceste reflexii perfecte (palindromuri) și afișează **cuvântul reflexie cu număr impar de litere și lungime maximă**. Dacă nu există niciunul, afișează mesajul **nici un simbol simetric**

**De exemplu**, pentru textul `radar civic noon level madam rotator`,  
se va afișa: `rotator`

## Limbajul Python

```
ins =input("Introd inscripția: ")
cuv =inscriptie.split()
rez =""
if cuv==cuv[::-1]and len(cuv)%2==1:
 if len(cuv) > len(rez): rez = cuv
if rez == "": print("niciun simbol simetric")
else: print(rez)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
bool Palindrom(const char cuv[]) {
 int n = strlen(cuv);
 for (int i = 0; i < n / 2; i++)
 if (cuv[i] != cuv[n-1-i]) return false;
 return true;
}

int main() {
 char linie[101];
 char celMaiLung[101] = "";
 int lung;
 cout << "Introdu inscripția ";
 cin.getline(linie, 101);
 char* cuvant = strtok(linie, " ");
 char celMaiLung[101] = "";
 while (cuvant != NULL) {
 lung=strlen(cuvant);
 if (Palindrom(cuvant)&& lungime % 2 == 1)
 if(lung > strlen(celMaiLung)) strcpy(celMaiLung, cuv);
 }
 cuvant = strtok(NULL, " ");
 if (strlen(celMaiLung) == 0) cout << "niciun simbol simetric";
 else cout << celMaiLung;
 }
 return 0;
}
```

38. Scrieți un program **Python/C++** care citește de la tastatură un text format din cel mult 100 de caractere, doar litere mici și mari ale alfabetului englez și spații. Literele din text sunt grupate în cuvinte iar cuvintele sunt separate prin câte un spațiu. Programul afișează pe ecran fiecare cuvânt precedat de forma sa inversă, în ordinea în care apar în șirul inițial.

**De exemplu,** pentru textul **Ana merge CuTaxi,**

se va afișa: **anA Ana egrem merge ixaTuC CuTaxi**

### Limbajul Python

```
text = input("Introduceți șirul: ").strip()
cuvinte = text.split()
for cuv in cuvinte:
 invers = cuv[::-1]
 print(invers, cuv, end=' ')
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;

void inverseaza(char cuv[], char invers[]) {
 int n = strlen(cuv);
 for (int i = 0; i < n; i++) invers[i] = cuv[n - 1 - i];
 invers[n] = '\0';
}

int main() {
 char linie[101], *p, cuvant[51], invers[51];
 cout << "Introduceți șirul: ";
 cin.getline(linie, 101);
 p = strtok(linie, " ");
 while (p != NULL) {
 strcpy(cuvant, p);
 inverseaza(cuvant, invers);
 cout << invers << " " << cuvant << " ";
 p = strtok(NULL, " ");
 }
 return 0;
}
```

39. Scrieți un program **Python/C++** care citește de la tastatură un șir format din cel mult 250 de caractere, doar litere mici ale alfabetului englez. Programul afișează numărul de vocale din text. Se consideră vocale literele: 'a', 'e', 'i', 'o', 'u'.

**De exemplu,** dacă se citește șirul **septembrie**, programul va afișa șirul **4**

### Limbajul Python

```
s=input()
v="aeiou"
nv=0
for i in range(len(s)):
 if s[i] in v:
 nv=nv+1
print(nv)
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char s[101];
 char voc[]="aeiou";
 int i,nv=0;
 cin.get(s,101);
 for(i=0; i<strlen(s);i++)
 if(strchr(voc, s[i]))
 nv++;
 cout<<nv;
 return 0;
}
```

40. Scrieți un program **Python/C++** care citește de la tastatură un text format din cel mult 100 de caractere, doar litere ale alfabetului englez și spații. Programul determină numărul de perechi de caractere consecutive identice din text.  
**De exemplu**, pentru textul citit `copiii vin in numar mareee la zoo` programul va afișa: 5.

#### Limbajul Python

```
s=input()
v="aeiou"
k = 0
for i in range(1,len(s)):
 if s[i]==s[i-1]: k+=1
print (k)
```

#### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{ char s[101];
 char voc[]="aeiou";
 int i,k=0;
 cin.get(s,101);
 for(i=1; i<strlen(s)-1;i++)
 if(s[i]==s[i+1]) k++;
 cout<<k;
 return 0;
}
```

41. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $0 < n < 100$ ) și  $n$  cuvinte. Un cuvânt este format din cel mult 31 de caractere, doar litere mici ale alfabetului englez. Programul afișează pe ecran cuvintele oglindite, în ordinea citirii lor.

**De exemplu**, pentru  $n=5$  și cuvintele:

```
ana
are
cojoc
rosu
frumos
programul va afișa:
ana
era
cojoc
usor
somurf.
```

## Limbajul Python

```
n = int(input())
for i in range(n):
 s = input().strip()
 caractere = list(s)
 for j in range(len(caractere) // 2):
 caractere[j], caractere[-1 - j] = caractere[-1 - j], caractere[j]
 oglindit = ''.join(caractere)
 print(oglindit)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char s[31], c;
 int i, j, n;
 cin >> n;

 for(i=1; i<=n; i++)
 {
 cin >> s;
 for(j=0; j<strlen(s)/2; j++)
 {
 c=s[j];
 s[j]=s[strlen(s)-1-j];
 s[strlen(s)-1-j]=c;
 }
 cout << s << endl;
 }
 return 0;
}
```

42. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $0 < n < 100$ ) și  $n$  cuvinte. Un cuvânt este format din cel mult 31 de caractere, doar litere mici ale alfabetului englez. Programul afișează pe ecran numărul de cuvintele palindrom, respectiv numărul de cuvinte care încep și se termină cu același caracter. Cele două numere se afișează pe rânduri diferite.

**De exemplu**, pentru  $n=5$  și cuvintele:

ana

are

cojoc

rosu

frumos programul va afișa:

2

2

## Limbajul Python

```
n = int(input())
np = 0 # număr de palindroame
nc = 0 # număr de cuvinte cu prima literă egală cu ultima
for i in range(n):
 s = input().strip()
 a = s
 # păstrăm originalul pentru comparație
 s_reversed = s[::-1]
 # inversăm șirul
 if a == s_reversed: np = np+1
 if s[0] == s[-1]: nc = nc+1
print(np)
print(nc)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char s[31],c,a[31];
 int i,j,n,np=0,nc=0;
 cin>>n;
 for(i=1; i<=n;i++)
 {
 cin>>s; strcpy(a,s);
 for(j=0;j<strlen(s)/2;j++)
 {
 c=s[j]; s[j]=s[strlen(s)-1-j]; s[strlen(s)-1-j]=c;
 }
 if(strcmp(a,s)==0) np++;
 if(s[0]==s[strlen(s)-1]) nc++;
 }
 cout<<np<<endl<<nc;
 return 0;
}
```

## VI. Liste

1. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul afișează ca în exemplu.

- a) lista, de la primul la ultimul element;
- b) lista, de la ultimul la primul element;
- c) doar elementele listei aflate pe poziții pare, de la primul la ultimul;
- d) doar elementele impare ale listei, de la primul la ultimul;

**Exemplu:** dacă se citesc, în această ordine, de pe linii separate, valorile:

5 2 8 0 9 90

se afișează pe ecran:

- a) Lista de la primul la ultimul element:  
[2, 8, 0, 9, 90]
- b) Lista de la ultimul la primul element:  
[90, 9, 0, 8, 2]
- c) Elementele de pe pozitii pare:  
[8, 9]
- d) Elementele impare din lista:  
[9]

### Limbajul Python

```
n = int(input())
lista = []
for i in range(n):
 numar = int(input())
 lista.append(numar)
print("\na) Lista de la primul la ultimul element:")
print(lista)
print("\nb) Lista de la ultimul la primul element:")
print(lista[::-1])
print("\nc) Elementele de pe pozitii pare:")
print([lista[i] for i in range(1, len(lista), 2)])
print("\nd) Elementele impare din lista:")
print([x for x in lista if x % 2 == 1])
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n;
 int lista[100];
 cin>>n;
 for(int i=0;i<n;++i)
 cin>>lista[i];
 cout<<"\na) Lista de la primul la ultimul element:";
 for(int i=0;i<n;++i)
 cout<<lista[i]<<" ";

 cout<<"\nb) Lista de la ultimul la primul element:";
 for(int i=n-1;i>=0;--i)
 cout<<lista[i]<<" ";
```

```

 cout <<"\n"); Elemente de pe pozitii pare: ";
 for(int i=1;i<n; i+=2)
 cout<<lista[i]<<" ";
 cout<<"\n"); Elemente impare din lista: ";
 for(int i=0;i<n;++i)
 if(lista[i]%2==1)
 cout<<lista[i]<<" ";
 return 0;
}

```

2. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul afișează lista, apoi adaugă un nou element egal cu suma celor  $n$  elemente și afișează lista.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 5 12 3 9 1 0

se afișează

Lista initiala:

[12, 3, 9, 1, 0]

Lista finala:

[12, 3, 9, 1, 0, 25]

### Limbajul Python

```

n = int(input())
lista = []
for i in range(n):
 numar = int(input())
 lista.append(numar)
print("\nLista initiala:")
print(lista)
suma = sum(lista)
lista.append(suma)
print("\nLista finala:")
print(lista)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 int n;
 int lista[101];
 cin >> n;
 for(int i=0;i<n;++i)
 cin>>lista[i];

 cout << "\nLista initiala: ";
 for(int i=0;i<n;++i)
 cout<<lista[i]<<" ";
 int suma=0;
 for(int i=0;i<n;++i)
 suma+=lista[i];
 lista[n]=suma;
 ++n;
 cout<<"\n\nLista finala: ";
 for(int i=0;i<n;++i)
 cout<<lista[i]<<" ";
 return 0;
}

```

3. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt întregi din intervalul  $[-1000, 1000]$ . Programul afișează lista, o sortează crescător, o sortează descrescător, șterge elementele cu ultima cifră 5 din lista inițială. După fiecare operație afișează listele obținute ca în exemplu.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 6 23 -135 55 -89 235 -21

se afișează

Lista initiala:

[23, -135, 55, -89, 235, -21]

Lista sortata crescator:

[-135, -89, -21, 23, 55, 235]

Lista sortata descrescator:

[235, 55, 23, -21, -89, -135]

Lista modificata:

[23, -89, -21]

### Limbajul Python

```
n = int(input())
lista = []
for i in range(n):
 numar = int(input())
 lista.append(numar)
print("\nLista initiala:")
print(lista)
lista_cresc = sorted(lista)
print("\nLista sortata crescator:")
print(lista_cresc)
lista_descresc = sorted(lista, reverse=True)
print("\nLista sortata descrescator:")
print(lista_descresc)
lista_5 = [x for x in lista if abs(x) % 10 != 5]
print("\nLista modificata:")
print(lista_5)
```

### Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
 int n, lista[100];
 cin >> n;
 for(int i=0; i<n; i++)
 cin >> lista[i];
 cout << "\nLista initiala:";
 for(int i=0; i<n; i++)
 cout << lista[i] << " ";
 int listaCresc[100];
 for(int i=0; i<n; i++)
 listaCresc[i] = lista[i];
 for(int i=0; i<n-1; i++)
 for(int j=i+1; j<n; j++)
 if(listaCresc[i] > listaCresc[j])
```

```

 {
 int temp=listaCresc[i];
 listaCresc[i]=listaCresc[j];
 listaCresc[j]=temp;
 }
 cout<<"\nLista sortata crescator:";
 for(int i=0;i<n; i++)
 cout<<listaCresc[i]<<" ";
 int listaDesc[100];
 for(int i=0; i<n; i++)
 listaDesc[i]=lista[i];
 for(int i=0; i<n-1; i++)
 for(int j=i+1; j<n; j++)
 if(listaDesc[i]<listaDesc[j])
 {
 int temp=listaDesc[i];
 listaDesc[i]=listaDesc[j];
 listaDesc[j]=temp;
 }
 cout<<"\nLista sortata descrescator:";
 for(int i=0; i<n; i++)
 cout<<listaDesc[i]<<" ";
 int lista_5[100];
 int m=0;
 for(int i=0;i<n;i++)
 if(abs(lista[i])%10!=5)
 {
 lista_5[m]=lista[i];
 m++;
 }
 cout<<"\nLista modificata:";
 for(int i=0;i<m;i++)
 cout<<lista_5[i]<<" ";
 return 0;
}

```

4. Scrieți un program **Python/C++** care citește de la tastatură două numere naturale nenule  $n_1$  și  $n_2$  ( $n_1, n_2 < 100$ ) și care construiește două liste **L1** și **L2** cu  $n_1$ , respectiv  $n_2$  elemente, numere naturale citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 100]$ . Programul afișează ca în exemplu cele două liste, apoi:

- 1) o nouă listă **L3**, cu elementele comune celor două liste **L1** și **L2**, scrise o singură dată;
- 2) o nouă listă **L4**, cu elementele comune și necomune celor două liste **L1** și **L2**, scrise o singură dată;
- 3) o nouă listă **L5**, cu elementele care apar doar în prima listă **L1** și nu apar în cea de-a doua listă **L2**, o singură dată;

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 5 4 23 12 45 67 19 67 78 45 12

se afișează

**L1:** [23, 12, 45, 67, 19]

**L2:** [67, 78, 45, 12]

**L3:** [67, 12, 45]

**L4:** [67, 12, 45, 78, 19, 23]

**L5:** [19, 23]

## Limbajul Python

```
n1 = int(input())
n2 = int(input())
L1 = []
for i in range(n1):
 x = int(input())
 L1.append(x)
L2 = []
for i in range(n2):
 x = int(input())
 L2.append(x)
print("\nL1:", L1)
print("L2:", L2)
L3 = list(set(L1) & set(L2))
print("L3:", L3)
L4 = list(set(L1) | set(L2))
print("L4:", L4)
L5 = list(set(L1) - set(L2))
print("L5:", L5)
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n1,n2;
 int L1[100],L2[100];
 int L3[100],L4[200],L5[100];
 int m3=0,m4=0,m5=0;
 cin>>n1>>n2;
 for(int i=0; i<n1; i++)
 cin>>L1[i];
 for(int i=0; i<n2; i++)
 cin>>L2[i];
 cout<<"\nL1:";
 for(int i=0; i<n1; i++)
 cout<<L1[i]<<" ";
 cout<<"\nL2:";
 for(int i=0; i<n2; i++)
 cout<<L2[i]<<" ";
 for(int i=0; i<n1; i++)
 for(int j=0; j<n2; j++)
 {
 if(L1[i]==L2[j])
 {
 bool exista=false;
 for(int k=0; k<m3; k++)
 {
 if(L3[k]==L1[i])
 {
 exista=true;
 break;
 }
 }
 if(!exista)
 L3[m3++]=L1[i];
 }
 }
}
```

```

cout<<"\nL3:";
for(int i=0; i<m3; i++) cout<<L3[i]<<" ";
for(int i=0; i<n1; i++)
{
 bool exista=false;
 for(int k=0; k<m4; k++)
 if(L4[k]==L1[i]) { exista=true; break; }
 if(!exista) L4[m4++]=L1[i];
}
for(int i=0; i<n2; i++)
{
 bool exista=false;
 for(int k=0; k<m4; k++)
 if(L4[k]==L2[i]) { exista=true; break; }

 if(!exista) L4[m4++]=L2[i];
}
cout<<"\nL4:";
for(int i=0;i<m4;i++) cout<<L4[i]<<" ";
for(int i=0;i<n1;i++)
{
 bool inL2=false;
 for(int j=0;j<n2;j++)
 if(L1[i]==L2[j])
 { inL2=true; break; }
 if(!inL2)
 {
 bool deja=false;
 for(int k=0;k<m5;k++)
 if(L5[k]==L1[i]) { deja=true; break; }
 if(!deja) L5[m5++]=L1[i];
 }
}
cout<<"\nL5:";
for(int i=0;i<m5;i++) cout<<L5[i]<<" ";
return 0;
}

```

5. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește două liste  $L1$  și  $L2$  cu  $n$  elemente, numere naturale citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 100]$ . Programul afișează ca în exemplu cele două liste, apoi:

- o nouă listă  $L3$ , cu elementele egale cu suma elementelor din listele  $L1$  și  $L2$  aflate pe poziții omoloage;
- o nouă listă  $L4$ , cu elementele egale cu cel mai mare divizor comun al elementelor din listele  $L1$  și  $L2$  aflate pe poziții omoloage;

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile

5 20 24 100 6 91 15 80 35 63 20

se afișează

$L1$ : 20 24 100 6, 91

$L2$ : 15 80 35 63 20

$L3$ : 35 104, 135 69, 111

$L4$ : 5 8 5 3 1

## Limbajul Python

```
n = int(input())
L1 = []
L2 = []
for i in range(n):
 x = int(input())
 L1.append(x)
for i in range(n):
 x = int(input())
 L2.append(x)
print("L1:", *L1)
print("L2:", *L2)
L3 = []
for i in range(n):
 L3.append(L1[i] + L2[i])
print("\nL3:", *L3)
L4 = []
for i in range(n):
 a = L1[i]
 b = L2[i]
 while b != 0:
 r = a % b
 a = b
 b = r
 L4.append(a)
print("\nL4:", *L4)
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;

int main()
{
 int n;
 int L1[100], L2[100], L3[100], L4[100];
 cin >> n;
 for(int i=0; i<n; i++) cin >> L1[i];
 for(int i=0; i<n; i++)
 cin >> L2[i];
 cout << "\nL1:";
 for(int i=0; i<n; i++) cout << L1[i] << " ";
 cout << "\nL2:";
 for(int i=0; i<n; i++)
 cout << L2[i] << " ";
 for(int i=0; i<n; i++)
 L3[i] = L1[i] + L2[i];
 cout << "\nL3:";
 for(int i=0; i<n; i++)
 cout << L3[i] << " ";
 for (int i=0; i<n; i++)
 {
 int a=L1[i], b=L2[i];
```

```

 while(b!=0)
 {
 int r=a%b;
 a=b;
 b=r;
 }
 L4[i]=a;
 }
 cout<<"\nL4:";
 for(int i=0;i<n;i++)
 cout<<L4[i] << " ";
 return 0;
}

```

6. Scrieți un program **Python/C++** inițializează o listă **L1** cu 7 elemente nume de concurenți, **L1 = ['Mihai', 'Gigi', 'Ana', 'Dan', 'Ion', 'Vlad', 'Mara']**. Programul construiește a doua listă **L2** cu 7 elemente, numere naturale de cel mult două cifre, citite de la tastatură reprezentând punctajele la un concurs. Programul afișează ca în exemplu, concurenții cu cel mai mare punctaj.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 23 50 56 23 2 56 56 se afișează

Ana  
Vlad  
Mara

### Limbajul Python

```

L1 = ['Mihai', 'Gigi', 'Ana', 'Dan', 'Ion', 'Vlad', 'Mara']
L2 = []
for i in range(7):
 p = int(input())
 L2.append(p)
max_punctaj = max(L2)
for i in range(7):
 if L2[i] == max_punctaj:
 print(L1[i])

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 char L1[7][10]={"Mihai","Gigi","Ana","Dan","Ion","Vlad","Mara"};
 int L2[7];
 for(int i=0;i<7;i++)
 cin>>L2[i];
 int max_punctaj = L2[0];
 for(int i=1;i<7;i++)
 if(L2[i]>max_punctaj)
 max_punctaj=L2[i];
 for(int i=0;i<7;i++)
 if(L2[i]==max_punctaj)
 cout<<L1[i]<<"\n";
 return 0;
}

```

7. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt întregi din intervalul  $[-1000, 1000]$ . Programul afișează lista, șterge toate elementele negative și apoi afișează lista modificată.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 5 -98 8 -6 -8 56

se afișează

-98 8 -6 -8 56

8 56

| Limbajul Python                                                                                                                                  | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n = int(input()) L = [] for _ in range(n):     x = int(input())     L.append(x) print(*L) L = [x for x in L if x &gt;= 0] print(*L) </pre> | <pre> #include &lt;iostream&gt; using namespace std; int main() {     int n;     int L[100];     int m=0;     cin&gt;&gt;n;     for(int i=0;i&lt;n;i++) cin&gt;&gt;L[i];     for(int i=0;i&lt;n;i++)         cout&lt;&lt;L[i]&lt;&lt;" ";     cout&lt;&lt;"\n";     for(int i=0;i&lt;n;i++)         if (L[i]&gt;=0)             L[m++]=L[i];     for(int i=0;i&lt;m;i++)         cout&lt;&lt;L[i]&lt;&lt;" ";     return 0; } </pre> |

8. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[0, 1000]$ . Programul afișează lista, inserează între oricare două numere alăturate diferența în valoare absolută și apoi afișează lista modificată.

**Exemplu:** dacă pentru  $n$  se citește 5 iar lista are valorile 2 45 31 60 50

se afișează 2 43 45 14 31 29 60 10 50

| Limbajul Python                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> n=int(input()) L=[] for _ in range(n): L.append(int(input())) print(*L) m=n i=n-1 while i&gt;0:     dif=abs(L[i]-L[i-1])     L.insert(i,dif)     i-=1     m+=1 print(*L) </pre> |

## Limbajul C/C++

```
#include<iostream>
using namespace std;
int main()
{ int n,L[200],m;
 cin>>n;
 for(int i=0;i<n;i++) cin>>L[i];
 for(int i=0;i<n;i++) cout<<L[i]<<" ";
 cout<<"\n";
 m=n;
 for(int i=n-1;i>0;i--)
 { int dif=L[i]>L[i-1]?L[i]-L[i-1]:L[i-1]-L[i];
 for(int j=m;j>i;j--) L[j]=L[j-1];
 L[i]=dif;
 m++;
 }
 for(int i=0;i<m;i++) cout<<L[i]<<" ";
 return 0;
}
```

9. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[0, 1000]$ . Programul afișează lista, modifică ordinea elementelor astfel încât cele cu prima cifră impară să ocupe primele poziții în ordine crescătoare iar celelalte să ocupe ultimele poziții în ordine descrescătoare și apoi afișează lista modificată.

**Exemplu:** dacă pentru  $n$  se citește 8 iar lista are valorile 123 789 45 209 35 683 49 56 se afișează

123 789 45 209 35 683 49 56

35 56 123 789 683 209 49 45

## Limbajul Python

```
n=int(input())
L=[]
for _ in range(n):
 L.append(int(input()))
print(*L)
impari=[]
pari=[]
for x in L:
 y=x
 while y>=10:
 y//=10
 if y%2==1:
 impari.append(x)
 else:
 pari.append(x)
for i in range(len(impari)-1):
 for j in range(i+1,len(impari)):
 if impari[i]>impari[j]:
 impari[i],impari[j]=impari[j],impari[i]
```

```

for i in range(len(pari)-1):
 for j in range(i+1,len(pari)):
 if pari[i]<pari[j]:
 pari[i],pari[j]=pari[j],pari[i]
L_mod=impari+pari
print(*L_mod)

```

## Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 int n,L[100];
 cin>>n;
 for(int i=0;i<n;i++) cin>>L[i];
 for(int i=0;i<n;i++) cout<<L[i]<<" ";
 cout<<"\n";
 int impari[100],par[100];
 int imp=0,p=0;
 for(int i=0;i<n;i++)
 {
 int x=L[i];
 while(x>=10)x/=10;
 if(x%2==1) impari[imp++]=L[i];
 else par[p++]=L[i];
 }
 for(int i=0;i<imp-1;i++)
 for(int j=i+1;j<imp;j++)
 if(impari[i]>impari[j])
 {
 int aux=impari[i];
 impari[i]=impari[j];
 impari[j]=aux;
 }
 for(int i=0;i<p-1;i++)
 for(int j=i+1;j<p;j++)
 if(par[i]<par[j])
 {
 int aux=par[i];
 par[i]=par[j];
 par[j]=aux;
 }
 for(int i=0;i<imp;i++) L[i]=impari[i];
 for(int i=0;i<p;i++) L[imp+i]=par[i];
 for(int i=0;i<n;i++) cout<<L[i]<<" ";
 return 0;
}

```

10. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[0, 1000]$ . Programul afișează lista, verifică dacă toate elementele sunt diferite două câte două și afișează un mesaj corespunzător (**da/nu**).

**Exemplu:** dacă pentru  $n$  se citește 5 iar lista are valorile 45 23 14 45 6

se afișează

45 23 14 45 6

nu

## Limbajul Python

```
n=int(input())
L=[]
for _ in range(n): L.append(int(input()))
print(*L)
distincte=True
for i in range(n-1):
 for j in range(i+1,n):
 if L[i]==L[j]: distincte=False
if distincte: print("da")
else: print("nu")
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n,L[100];
 bool distincte=true;
 cin>>n;
 for(int i=0;i<n;i++)
 cin>>L[i];
 for(int i=0;i<n;i++)
 cout<<L[i]<<" ";
 cout<<"\n";
 for(int i=0;i<n-1;i++)
 for(int j=i+1;j<n;j++)
 if(L[i]==L[j])
 distincte=false;
 if(distincte) cout<<"da";
 else cout<<"nu";
 return 0;
}
```

11. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură în ordine crescătoare, apoi un număr natural nenul  $m$  ( $m < 100$ ) și care construiește o a doua listă cu numere citite de la tastatură în ordine crescătoare. Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul construiește în memorie o a treia listă cu elementele aranjate în ordine crescătoare apoi o afișează pe ecran.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile

5 2 4 5 7 9

4 1 4 6 12

Se va afișa lista: 1 2 4 4 5 6 9 12

## Limbajul Python

```
n1=int(input("Numărul de elemente din prima listă: "))
print("Introduce elementele:")
lista1=[]
for _ in range(n1):
 lista1.append(int(input()))
```

```

n2=int(input("Numărul de elemente din a doua listă: "))
print("Introduce elementele:")
lista2=[]
for _ in range(n2):
 lista2.append(int(input()))
i=0
j=0
n3=0
lista3 = []
while i<n1 and j<n2:
 if lista1[i] < lista2[j]:
 lista3.append(lista1[i])
 i+= 1
 else:
 lista3.append(lista2[j])
 j+= 1
 n3+=1
while i<n1:
 lista3.append(lista1[i])
 i+= 1
 n3+= 1
while j<n2:
 lista3.append(lista2[j])
 j+=1
 n3+=1
print(lista3)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main() {
 int n1,n2,i,j,lista1[101],lista2[101],lista3[201],n3=0;
 cout<<"Numărul de elemente din prima listă: ";
 cin>>n1;
 cout<<"Introduce elementele:";
 for(i=0;i<n1;i++) cin>>lista1[i];
 cout<<"Numărul de elemente din a doua listă: ";
 cin>>n2;
 cout<<"Introduce elementele:";
 for(i=0;i<n2;i++)cin>>lista2[i];
 i=0; j=0;
 while(i<n1 && j<n2) {
 if(lista1[i]<lista2[j])
 { lista3[n3]=lista1[i]; i++; n3++; }
 else { lista3[n3]=lista2[j]; j++; n3++; }
 }
 while(i<n1)
 { lista3[n3]=lista1[i]; i++; n3++; }
 while(j<n2)
 { lista3[n3]=lista2[j]; j++; n3++; }
 for(i=0;i<n3;i++) cout<<lista3[i]<<" ";
 return 0;
}

```

12. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 1000]$ . Programul construiește în memorie o listă în care se regăsesc mai întâi toate elementele pare apoi cele impare. Lista se afișează pe ecran.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 5 2 4 5 7 9 se va afișa 4 2 5 7 9

#### Limbajul Python

```
n=int(input("Numărul de elemente: "))
v=[]
for _ in range(n):
 x = int(input())
 if x%2==0:
 v.insert(0, x)
 else: v.append(x)
for elem in v: print(elem, end=' ')
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int v[101], i, j, n, x, nr=-1;
 cin >> n;
 for(int i=0; i<n; i++)
 {
 cin >> x;
 if(x%2==0)
 {
 nr++;
 for(j=nr; j>0; j--) v[j]=v[j-1];
 v[0]=x;
 }
 else { nr++; v[nr]=x; }
 }
 for(i=0; i<=nr; i++) cout << v[i] << " ";
 return 0;
}
```

13. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și un șir de  $n$  numere naturale din intervalul  $[2, 1000]$ . Să se insereze după fiecare pătrat perfect radicalul pătratului perfect. Lista se afișează pe ecran.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile

5 3 25 16 7 2

Se va afișa 3 25 5 16 4 7 2

#### Limbajul Python

```
n=int(input())
v=[]
for _ in range(n): v.append(int(input()))
i=0
while i<n:
 c=int(v[i]**0.5)
 if c*c==v[i]:
 n+=1
 v.insert(i+1, c)
 i+=1
 i+=1
```

```
print("Rezultatul final:")
for x in v: print(x, end=' ')
```

### Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{ int v[101],i,j,n,c;
 cin>>n;
 for(i=0;i<n;i++) cin>>v[i];
 for(i=0;i<n;i++)
 {
 c=sqrt(v[i]);
 if(c*c==v[i])
 { n++;
 for(j=n-1;j>i;j--) v[j]=v[j-1];
 v[i+1]=c;i++;
 }
 }
 for(i=0;i<n;i++) cout<<v[i]<<" ";
 return 0;
}
```

14. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și un șir de  $n$  numere naturale din intervalul  $[2, 1000]$ . Să afișeze toate permutările circulare, fiecare permutare pe câte un rand, elementele fiind separate printr-un spațiu.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 4 3 4 5 2

Se va afișa:

```
3 4 5 2
4 5 2 3
5 2 3 4
2 3 4 5
```

### Limbajul Python

```
n=int(input())
v=[]
for _ in range(n):
 v.append(int(input()))
for _ in range(n):
 for elem in v:
 print(elem, end=' ')
 print()
 aux=v[0]
 for j in range(n-1):
 v[j]=v[j+1]
 v[n-1]=aux
```

### Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{ int v[101],i,j,aux,n;
 cin>>n;
 for(i=0;i<n;i++) cin>>v[i];
 for(i=0;i<n;i++)
 { for(j=0;j<n;j++)
 cout<<v[j]<<" ";
 cout<<endl;
 aux=v[0];
 for(j=0;j<n-1;j++)
 v[j]=v[j+1];
 v[n-1]=aux;
 }
 return 0;
}
```

15. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $2 < n < 100$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 1000]$ . Să afișeze toate tripletele de câte trei elemente, fiecare triplet pe câte un rand, păstrându-se ordinea numerelor din șirul inițial, elementele fiind separate printr-un spațiu, ca în exemplu.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile 4 3 4 5 2

Se va afișa

3 4 5

3 4 2

3 5 2

4 5 2

### Limbajul Python

```
n=int(input())
v=list(map(int, input().split()))
for i in range(n - 2):
 for j in range(i + 1, n - 1):
 for k in range(j + 1, n): print(v[i], v[j], v[k])
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n,i,j,k,v[101];
 cin>>n;
 for(i=0; i<n; i++) cin>>v[i];
 for(i=0;i<n-2;i++)
 for(j=i+1;j<n-1;j++)
 for(k=j+1;k<n;k++)
 cout<<v[i]<<" "<<v[j]<<" "<<v[k]<<"\n";
 return 0;}
```

16. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $2 < n < 100$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 1000]$  ce reprezintă numerele de pe tricourile unor elevi. Să se elimine din 3 în 3 până rămâne un singur elev și să se afișeze pe ecran numerele elevilor rămași după fiecare eliminare, cu spații între ele și pe linii separate.

**Exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile

8 7 3 4 5 2 7 6 1

Se va afișa

7 3 4 5 2 7 6 1

7 3 5 2 7 6 1

7 3 5 2 6 1

3 5 2 6 1

3 5 6 1

5 6 1

5 6

6

## Limbajul Python

```
n=int(input())
v=list(map(int, input().split()))
k=0
i=0
while n!=1:
 k+=1
 if k%3==0:
 v.pop(i)
 n-=1
 i-=1
 for elem in v: print(elem, end=' ')
 print()
 i+=1
 if i>n-1: i=0
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n,i,j,v[101],k=0;
 cin>>n;
 for(i=0; i<n; i++) cin>>v[i];
 i=0;
 while(n!=1)
 {
 k++;
 if(k%3==0)
 {
 for(j=i;j<n-1;j++) v[j]=v[j+1];
 n--;
 i--;
 for(j=0;j<n;j++) cout<<v[j]<<" ";
 cout<<endl;
 }
 i++;
 if(i>n-1) i=0;
 }
 return 0;
}
```

17. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $2 < n < 100$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 1000]$ . Programul creează o listă cu cele  $n$  numere, apoi, elimină din listă fiecare număr impar și dublează în listă fiecare număr par, apoi afișează lista pe o linie a ecranului.

**Exemplu:** dacă se citesc, în această ordine: 8

7 3 4 5 2 7 6 1

Programul va afișa lista:

4 4 2 2 6 6

## Limbajul Python

```
n = int(input("n= "))
v = list(map(int, input("sir numere: ").split()))

i = 0
while any(x % 2 != 0 for x in v): # cât timp există numere impare
 if v[i] % 2 == 0:
 v.insert(i + 1, v[i]) # dublare nr par
 i += 2
 else:
 v.pop(i) # eliminare nr impar
print("Lista finală:", v)
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n, v[2001], i, j;
 cout << "n= "; cin >> n;
 cout << "sirul de nr: ";
 for (i = 0; i < n; i++) cin >> v[i];
 i=0;
 while (i < n)
 { if (v[i] % 2 == 0)
 { for (j = n-1; j >= i; j--) v[j+1] = v[j];
 n++; i=i+2; }
 else
 { for (j = i; j < n - 1; j++) v[j] = v[j + 1];
 n--;
 }
 }
 for (int i = 0; i < n; i++)
 cout << v[i] << " ";
 cout << endl;
 return 0;
}
```

18. Scrieți un program **Python/C++** care citește de la tastatură un text (format din maxim 1000 de caractere), în care propozițiile sunt separate prin caracterul \. '. Programul construiește o listă care în care fiecare element memorează câte o propoziție, afișează prima propoziție din text și cea mai lungă propoziție din text, apoi sortează lexicografic elementele listei și afișează lista sortată, ca în exemplu. Propozițiile au lungimi diferite.

De exemplu, pentru textul:

**Vacanta mare. Ana are mere. La munte si la mare.**

programul afișează pe ecran, pe linii separate:

**Vacanta mare**

**La munte si la mare**

**['Ana are mere', 'La munte si la mare', 'Vacanta mare']**

## Limbajul Python

```
text = input("text: ")
lista = text.split('.')
Lista_p = [p.strip() for p in lista if p.strip() != '']
prop1 = Lista_p[0]
print(prop1)
pmax = max(Lista_p, key=len)
print(pmax)
Lista_s = sorted(Lista_p)
print(Lista_s)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;

int main() {
 char text[1001], pmax[256], aux[256];
 char Lista_p[1001][256];
 int n = 0, maxl=0, i, j;
 cout<<"text: ";
 cin.getline(text, 1001);
 char* p = strtok(text, ".");
 while (p)
 {
 while (p[0] == ' ') p++;
 strcpy(Lista_p[n], p); n++;
 if(strlen(p)>maxl)
 { maxl=strlen(p); strcpy(pmax, p); }
 p=strtok(NULL, ".");
 }
 cout << Lista_p[0]<<endl;
 cout << pmax<<endl;
 for(i=0; i<n-1; i++)
 for(j=i+1; j<n; j++)
 if(strcmp(Lista_p[i], Lista_p[j])>0)
 swap(Lista_p[i], Lista_p[j]);
 cout <<"["<< Lista_p[0];
 for (int i = 1; i < n; i++)
 cout <<" , "<< Lista_p[i];
 cout<<"]";
 return 0;
}
```

19. Scrieți un program **Python/C++** care citește de la tastatură un număr (format din maxim 1000 de cifre). Programul construiește o listă în care fiecare element memorează câte o cifră, apoi construiește o nouă listă care să conțină cifrele distincte din numărul citit, împreună cu numărul de apariții ale fiecărei cifre, ca în exemplu. Programul afișează lista creată pe ecran, ca în exemplu.

De exemplu, pentru numărul 7271231354999, programul va afișa lista:

[1, 2, 2, 2, 3, 2, 4, 1, 5, 1, 7, 2, 9, 3]

### Limbajul Python

```
n = input("n= ")
ap = [0] * 10
for c in n:
 ap[int(c)] += 1
Lista = []
for c in range(10):
 if ap[c] > 0:
 Lista.append(c)
 Lista.append(ap[c])
print(Lista)
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
struct element
{ int c, ap;}Lista[2000];
int main()
{ int k=0, ap[10]={0}, c,i;
 char n[1001];
 cout<<"n= "; cin>>n;
 for(i=0; i<strlen(n); i++)
 ap[n[i]-'0']++;
 for(c=0;c<=9;c++)
 if(ap[c])
 { Lista[k].c=c; Lista[k].ap=ap[c]; k++;}
 cout<<"["<<Lista[0].c<<" ", "<<Lista[0].ap;
 for(i=1; i<k; i++) cout<<" ", "<<Lista[i].c<<" ", "<<Lista[i].ap;
 cout<<"]";
 return 0;
}
```

20. Un text (format din maxim 1000 de caractere), conține mai multe perechi de câte două cuvinte de forma. Cele două cuvintele sunt separate printr-un spațiu, iar perechile sunt separate prin caracterul '#'. Cuvintele sunt formate din cel mult 20 de litere și încep cu o literă mare. În cadrul fiecărei perechi, cele două cuvinte reprezintă numele și prenumele unei persoane, ca în exemplu.

Scrieți un program **Python/C++** care citește de la tastatură textul, construiește o listă ale carei elemente memorează câte o pereche și înlocuiește prenumele fiecărei persoane cu inițială corespunzătoare urmată de caracterul \'. Programul afișează pe ecran lista modificată, ca în exemplu.

De exemplu, pentru textul: **Vasile Dan#Popa Mihai#Ion Ana#Popescu Maria**  
programul afișează lista: ['Vasile D.', 'Popa M.', 'Ion A.', 'Popescu M.']

### Limbajul Python

```
text = input("text: ")
lista_p = text.split('#')
lista = []
for p in lista_p:
 nume, pren = p.split()
 elem=nume+' '+pren[0] + ' .'
 lista.append(elem)
print(lista)
```

## Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{ char text[1001], *p, *t;
 int k=0, poz,i;
 char lista[1001][21], elem[41];
 cout<<"text= "; cin.getline(text,1001);
 p=strtok(text,"#");
 while(p)
 { strcpy(elem,p);
 t=strchr(elem,' ');
 poz=t-elem;
 elem[poz+2]='.';
 elem[poz+3]='\0';
 strcpy(lista[k],elem); k++;
 p=strtok(NULL,"#");
 }
 cout<<"["<<lista[0];
 for(i=1; i<k; i++)
 cout<<" "<<lista[i];
 cout<<"]";
 return 0;
}
```

21. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură, de pe linii separate. Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul afișează, ca în exemplu:

- doar ultima cifră a elementelor pare din listă, de la primul la ultimul;
- doar elementele impare din listei, de la primul la ultimul.

**Exemplu:** dacă se citesc, în această ordine, de pe linii separate, valorile:

```
5
61
32
3
64
5
```

se afișează pe ecran:

- a) Ultima cifră a numerelor pare din lista:  
2 4  
b) Elementele impare din lista:  
61 3 5

## Limbajul Python

```
n = int(input())
par = []
impar = []
for i in range(n):
 a = int(input())
 if a % 2 == 0: par.append(a%10)
 else: impar.append(a)
print("a) Ultima cifra a elementelor impare pare din lista:")
```

```
print(" ".join(map(str, par)))
print("b) Elementele impare din lista:")
print(" ".join(map(str, impar)))
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n,a,np=0,ni=0;
 int par[100],impar[100];
 cin>>n;
 for(int i=0;i<n;++i)
 { cin>>a;
 if(a%2==0)
 { par[np]=a; np++;}
 else
 { impar[ni]=a; ni++;}
 }
 cout <<"c) Ultima cifra a elementelor pare din lista:";
 for(int i=0;i<np; i++)
 cout<<par[i]<<" ";
 cout<<"\nd) Elementele impare din lista:";
 for(int i=0;i<ni;++i) cout<<impar[i]<<" ";
 return 0;
}
```

22. Scrieți un program **Python/C++** care citească de la tastatură două numere naturale nenule  $n$  ( $n < 100$ ) și  $k$  ( $k < 11$ ) și care construiește o listă cu numerele citite de la tastatură care pot fi scrise în baza  $k$ . Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul afișează lista creată, apoi adaugă un nou element egal cu numărul elementelor listei  $n$ , apoi se afișează lista creată.

**Exemplu:** dacă se citească, în această ordine, separate prin enter, valorile 5 4

31 17 23 10 2002

se afișează:

Lista initiala:

[31, 23, 10, 2002]

Lista finala:

[31, 23, 10, 2002, 4]

### Limbajul Python

```
n = int(input())
k = int(input())
nr=0
bazak = []
for i in range(n):
 a = int(input())
 Max=0
 ca=a
 while ca>0:
 if ca%10>Max: Max=ca%10
 ca=ca//10
 if Max<k:
 nr=nr+1
 bazak.append(a)
print("\nLista initiala:")
```

```

print(bazak)
bazak.append(nr)
print("\nLista finala:")
print(bazak)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 int n,k,a,nr=0;
 int bazak[101];
 cin >> n>>k;
 for(int i=0; i<n; ++i)
 {
 cin>>a;
 int Max=0; int ca=a;
 while(ca)
 { if(ca%10>Max)Max=ca%10;
 ca=ca/10;
 }
 if(Max<k)
 { bazak[nr]=a; nr++;}
 }
 cout << "\nLista initiala: ";
 for(int i=0; i<nr; ++i) cout<<bazak[i]<< " ";
 nr++;
 bazak[nr-1]=nr-1;
 cout<<"\nnLista finala:";
 for(int i=0; i<nr; ++i) cout<<bazak[i]<<" ";
 return 0;
}

```

23. Un cavaler a descoperit în timpul unei expediții, mai multe comori. Le-a evaluat și a marcat pe fiecare comoară un număr natural reprezentând valoarea ei. El a observat că, comorile marcate cu un număr par sunt niște comori magice. La întoarcerea din expediție, înainte de a preda regelui comorile, cavalerul observă că valoarea ultimei comori magice s-a dublat.

Scrieți un program în **Python/C++** care citește de la tastatură un număr natural nenul **n** ( $n < 1000$ ) și apoi o listă de **n** numere naturale (formate fiecare din cel mult 4 cifre), reprezentând valorile comorilor găsite. Programul va afișa lista valorilor comorilor predate regelui în care valoarea ultimei comori magice este dublată.

Exemplu: dacă se citesc, în această ordine valorile: 5 5 8 13 4 9, se afișează:

5 8 13 8 9

### Limbajul Python

```

n=int(input())
comori = list(map(int, input("Introduceți comorile: ").split()))
Parcurgem lista de la sfârșit spre început
for i in range(len(comori) - 1, -1, -1):
 if comori[i] % 2 == 0:
 comori[i] *= 2 # dublăm valoarea
 break
Afișăm lista modificată
print("Lista actualizată:", *comori)

```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, x, comori[1001], i;
 cout << "Introdu numarul de elemente: "; cin >> n;
 cout << "Introdu valorile: ";
 for (i = 0; i < n; i++) cin >> comori[i];
 for (i = n - 1; i >= 0; i--)
 if (comori[i] % 2 == 0) {
 comori[i] *= 2; break; }
 cout << "Lista actualizata: ";
 for (int i = 0; i < n; i++) cout << comori[i] << " ";
 return 0;
}
```

24. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $n < 100$ ), și apoi, de pe aceeași linie, separate prin câte un spațiu,  $n$  numere naturale care formează o listă de pietre magice.

Fiecare piatră este numerotată și este analizată în funcție de **suma cifrelor sale**;

- dacă suma cifrelor este **pară**, piatra este adăugată într-o listă de echilibru.
- dacă suma cifrelor este **impară**, piatra este adăugată într-o listă de haos.

După completarea celor două liste, programul afișează: elementele din lista de, sortate crescător, apoi, elementele din lista de haos, sortate tot crescător, toate pe o singură linie, separate prin spațiu.

Exemplu: dacă se citesc, în această ordine, separate prin enter, valorile:

6

13 24 18 37 40 12, se afișează: 13 24 37 40 12 18

## Limbajul Python

```
def suma_cifre(n):
 suma = 0
 while n > 0:
 suma += n % 10
 n //= 10
 return suma

n = int(input())
pietre = list(map(int, input("Introdu pietrele: ").split()))
echilibru = []
haos = []
for piatra in pietre:
 if suma_cifre(piatra) % 2 == 0:
 echilibru.append(piatra)
 else:
 haos.append(piatra)
echilibru.sort()
haos.sort()
rezultat = echilibru + haos
print(" ".join(map(str, rezultat)))
```

## Limbajul C/C++

```
#include <iostream>
#include <algorithm>
using namespace std;

int suma_cifre(int n) {
 int s = 0;
 while (n) {
 s += n % 10;
 n /= 10;
 }
 return s;
}

int main() {
 int n, x, echilibru[100], haos[100];
 int ne = 0, nh = 0;

 cin >> n;
 for (int i = 0; i < n; ++i) {
 cin >> x;
 int s = suma_cifre(x);
 if (s % 2 == 0)
 echilibru[ne++] = x;
 else
 haos[nh++] = x;
 }
 sort(echilibru, echilibru + ne);
 sort(haos, haos + nh);
 for (int i = 0; i < ne; ++i) cout << echilibru[i] << " ";
 for (int i = 0; i < nh; ++i) cout << haos[i] << " ";
 return 0;
}
```

25. Scrieți un program Python/C++ care citește de la tastatură un număr natural  $n$  ( $n \leq 100$ ), apoi  $n$  numere naturale reprezentând **coduri gravate pe pietrele oracolului**. Se cere să construiți două liste:

- una care conține **doar codurile ce sunt numere prime**,
- alta cu **cele care NU sunt prime, dar au cel puțin 4 divizori**.

Programul afișează mai întâi lista numerelor prime, apoi lista numerelor compuse cu cel puțin 4 divizori, fiecare pe o linie separată.

Exemplu: dacă se citesc, în această ordine, separate prin enter, valorile:

7

11 12 6 7 13 20 21

Se va afișa :

11 7 13

12 6 20 21

## Limbajul Python

```
def este_prim(x):
 if x < 2: return False
 for d in range(2, int(x**0.5) + 1):
 if x % d == 0: return False
 return True
```

```

def numar_divizori(x):
 cnt = 0
 for d in range(1, x + 1):
 if x % d == 0:
 cnt += 1
 return cnt
n = int(input())
L = []
for _ in range(n): L.append(int(input()))
prime = []
compuse = []
for x in L:
 if este_prim(x):
 prime.append(x)
 elif numar_divizori(x) >= 4:
 compuse.append(x)

print(*prime)
print(*compuse)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;

bool estePrim(int x) {
 if (x < 2) return false;
 if (x==2) return true;
 if(x%2==0) return false;
 for (int d = 3; d * d <= x; d+=2)
 if (x % d == 0) return false;
 return true;
}

int numarDivizori(int x) {
 int cnt = 0;
 for (int d = 1; d <= x; d++)
 if (x % d == 0) cnt++;
 return cnt;
}

int main() {
 int n, x, v[100];
 cin >> n;
 for (int i = 0; i < n; i++) cin >> v[i];
 for (int i = 0; i < n; i++)
 if (estePrim(v[i])) cout << v[i] << " ";
 cout << endl;
 // Afișăm numerele compuse cu cel puțin 4 divizori
 for (int i = 0; i < n; i++)
 if (!estePrim(v[i]) && numarDivizori(v[i]) >= 4)
 cout << v[i] << " ";
 return 0;
}

```

26. Scrieți un program Python/C++ care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ), apoi o listă de  $n$  numere naturale reprezentând forța unor zmei. Un zmeu este considerat „norocos” dacă forța lui este un număr palindrom (ex: 121, 44, 777). Programul va modifica fiecare valoare norocoasă din listă, marind-o cu suma cifrelor

eiș apoi va afisa, pe ecran, lista modificată.

Exemplu: dacă se citesc, în această ordine, separate prin enter, valorile: 5 121 32 77 89 44, programul va afișa : 127 32 91 89 52

### Limbajul Python

```
def este_palindrom(n):
 return str(n) == str(n)[::-1]

def suma_cifre(n):
 return sum(int(c) for c in str(n))

Citire date
n = int(input())
lista = []
for _ in range(n):
 lista.append(int(input()))

Procesare listă
for i in range(n):
 if este_palindrom(lista[i]):
 lista[i] += suma_cifre(lista[i])

Afișare rezultat
print(*lista)
```

### Limbajul C/C++

```
#include <iostream>
#include <vector>
using namespace std;

bool este_palindrom(int n) {
 int original = n, invers = 0;
 while (n > 0) {
 invers = invers * 10 + n % 10; n /= 10; }
 return original == invers;
}

int suma_cifre(int n) {
 int suma = 0;
 while (n > 0) {
 suma += n % 10; n /= 10; }
 return suma;
}

int main() {
 int n, x, i;
 cin >> n;
 int lista[101];
 for (i = 0; i < n; i++) cin >> lista[i];
 for (i = 0; i < n; i++)
 if (este_palindrom(lista[i]))
 lista[i] += suma_cifre(lista[i]);
}
```

```

 for (i = 0; i < n; i++) cout << lista[i] << " ";
 return 0;
}

```

27. Scrieți un program **Python/C++** care citește un număr natural nenul  $n$  ( $n < 100$ ), apoi o listă de  $n$  numere întregi. Verificați pentru fiecare număr din listă dacă diferența dintre prima și ultima cifră este 1. Înlocuiți aceste elemente cu "1!". Afișați lista modificată.

Exemplu: dacă se citesc, în această ordine, separate prin enter, valorile : 4 12 34 45 23, se va afișa : 1! 34 45 1!

### Limbajul Python

```

def prima_cifra(n):
 while n >= 10: n //= 10
 return n

def ultima_cifra(n):
 return n % 10

n = int(input())
lista = []
for _ in range(n): lista.append(int(input()))
for i in range(n):
 prima = prima_cifra(lista[i])
 ultima = ultima_cifra(lista[i])
 if abs(prima - ultima) == 1: lista[i] = "1!"

print(*lista)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main() {
 int n;
 cin >> n;
 int lista[100];
 for (int i = 0; i < n; i++) cin >> lista[i];
 for (int i = 0; i < n; i++) {
 int x = lista[i];
 int ultima = x % 10;
 int prima = x;
 while (prima >= 10) prima /= 10;
 if (abs(prima - ultima) == 1)
 cout << "1! ";
 else cout << x << " ";
 }
 return 0;
}

```

28. Scrieți un program Python/C++ care citește un număr natural nenul  $n$  ( $n < 100$ ), apoi o listă de  $n$  numere naturale. Programul va determina înlocuirea fiecărui număr care conține doar cifrele 0, 1, 6, 8 sau 9, cu oglinditul său, apoi va afișa, pe ecran, lista modificată.

Exemplu: dacă se citesc, în această ordine, separate prin enter, numerele: 5 101 123 86 609 45, programul va afișa : 101 123 68 906 45

### Limbajul Python

```
def are_cifre_valide(n):
 for cifra in str(n):
 if cifra not in "01689":
 return False
 return True

def inverseaza(n):
 return int(str(n)[::-1])

n = int(input())
lista = []
for _ in range(n):
 lista.append(int(input()))
for i in range(n):
 if are_cifre_valide(lista[i]):
 lista[i] = inverseaza(lista[i])
print(*lista)
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
bool are_cifre_valide(int n) {
 while (n > 0) {
 int cifra = n % 10;
 if (cifra != 0 && cifra != 1 && cifra != 6
 && cifra != 8 && cifra != 9) return false;
 n /= 10;
 }
 return true;
}
int inverseaza(int n) {
 int invers = 0;
 while (n > 0) { invers = invers * 10 + n % 10; n /= 10; }
 return invers;
}
int main() {
 int n, x, lista[100];
 cin >> n;
 for (int i = 0; i < n; i++) cin >> lista[i];
 for (int i = 0; i < n; i++)
 if (are_cifre_valide(lista[i]))
 lista[i] = inverseaza(lista[i]);
 for (int i = 0; i < n; i++) cout << lista[i] << " ";
 return 0;
}
```

29. Scrieți un program **Python/C++** care citește un număr natural nenul  $n$  ( $n < 100$ ), apoi o listă de  $n$  numere naturale. Programul determină înlocuirea fiecărui număr format doar din cifre prime cu produsul cifrelor sale, apoi afișează pe ecran lista modificată. Exemplu: dacă se citesc, în această ordine, separate prin enter, valorile : 6 235 54 357 23 71 444, programul va afișa : 30 54 105 6 71 444

#### Limbajul Python

```
def are_cifre_prime(n):
 cifre_prime = {'2', '3', '5', '7'}
 return all(c in cifre_prime for c in str(n))

def produs_cifre(n):
 produs = 1
 for c in str(n): produs *= int(c)
 return produs

n = int(input())
lista = []
for _ in range(n): lista.append(int(input()))
for i in range(n): if are_cifre_prime(lista[i]):
 lista[i] = produs_cifre(lista[i])
print(*lista)
```

#### Limbajul C/C++

```
#include <iostream>
using namespace std;
bool are_cifre_prime(int n) {
 while (n > 0) {
 int cif = n % 10;
 if (cif!=2 && cif!=3 && cif!=5 && cif!=7) return false;
 n /= 10; }
 return true;
}
int produs_cifre(int n) {
 int produs = 1;
 while (n > 0) { produs *= n % 10; n /= 10; }
 return produs;
}
int main() {
 int n, lista[100];
 cin >> n;
 for (int i = 0; i < n; i++) cin >> lista[i];
 for (int i = 0; i < n; i++)
 if (are_cifre_prime(lista[i]))
 lista[i] = produs_cifre(lista[i]);
 for (int i = 0; i < n; i++) cout << lista[i] << " ";
 return 0; }
```

30. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $2 < n < 35$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 10]$ , reprezentând rezultatele unei evaluări la informatică a unei clase cu  $n$  elevi. Raportat la datele

evaluării, programul afișează pe ecran, pe linii separate, următoarele:

- a) **ma**, reprezentând media clasei
- b) **nr**, reprezentând numărul elevilor din clasă care au note strict mai mari decât media clasei
- c) mesajul **omogen**, dacă numărul elevilor clasei care au obținut note din intervalul  $[|ma| - 2, |ma| + 2]$  reprezintă cel puțin 80% din **n**, respectiv mesajul **eterogen**, în caz contrar.

**Exemplu:** dacă se citesc, în această ordine, separate prin **enter**, respectiv prin câte un spațiu, valorile

10  
6 9 8 5 10 9 3 7 7 8

Se va afișa

7.2

5

omogen

iar dacă se citesc, respectând același format de citire a datelor, valorile

10  
6 9 8 5 10 9 3 7 2 2

Se va afișa

6.1

5

eterogen

### Limbajul Python

```
n=int(input("Numarul de elevi din clasa, 2<n<35: "))
v=list(map(int, input("Lista notelor, nr. nat. din [1,10]:").split()))
nr,cati=0,0
ma=sum(v)/n
print(ma)
for x in v:
 if x>ma: nr+=1
 if x>=int(ma)-2 and x<=int(ma)+2: cati+=1
if cati>=0.8*n: print("omogen")
else: print("eterogen")
print(nr)
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n,v[34],i,nr=0,cati=0;
 float ma,s=0;

 cout<<"Numarul de elevi din clasa (2<n<35): "; cin>>n;
 cout<<"Lista notelor (nr. nat. din [1,10] separate prin spatiu):";
 for(i=0;i<n;i++) {cin>>v[i]; s+=v[i];}
 ma=s/n;
 for(i=0;i<n;i++)
 {if(v[i]>ma) nr++;
 if(v[i]>=int(ma)-2 and v[i]<=int(ma)+2) cati++;
 }
 cout<<ma<<"\n"<<nr<<"\n";
 if(cati>=0.8*n) cout<<"omogen";
 else cout<<"eterogen";
 return 0;
}
```

31. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $2 < n \leq 100$ ) și un șir de  $n$  numere naturale, având toate cifrele nenule, din intervalul  $[1, 100000000)$ , în care cel puțin un element are un număr par de cifre. Programul construiește o listă cu cele  $n$  numere, apoi, elimină din listă fiecare element cu număr impar de cifre și transformă restul elementelor astfel: interschimbă prima jumătate din număr, cu a doua jumătate din număr. Programul afișează noua listă pe o linie a ecranului, ca în exemplu.

**Exemplu:** dacă se citesc, în această ordine, separate prin **enter**, respectiv prin câte un spațiu, valorile

4

12 123 5678 212688

Programul va afișa lista:

Noua listă: 21 7856 688212

### Limbajul Python

```
n=int(input())
v=list(map(int, input().split()))
i=0
while i<n:
 x=str(v[i])
 l=len(x)
 if(l%2!=0):
 v.pop(i)
 n-=1
 else:
 y=x[:l//2]
 z=x[l//2:]
 v[i]=z+y
 i+=1
print("Noua lista:", *v)
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;
int main()
{
 int n,v[100],i,j;
 char s[10];
 cin>>n;
 for(i=0;i<n;i++)
 cin>>v[i];
 for(i=0;i<n;i++)
 {
 itoa(v[i],s,10);
 int l=strlen(s);
 if(l%2!=0)
 {
 for(j=i;j<n-1;j++) v[j]=v[j+1];
 i--;
 n--;
 }
 else
 {
 char a[10]="",b[10]="";
 strncpy(a,s,l/2);
 strcpy(b,s+l/2);
 strcpy(s,b);strcat(s,a);
 }
 }
}
```

```

 v[i]=atoi(s);
 }
}
cout<<"Noua lista: ";
for(i=0;i<n;i++) cout<<v[i]<<' ';
return 0;
}

```

32. Un șir de numere naturale, având minim 3 elemente, se numește șir **munte** dacă elementele din șir sunt în ordine strict crescătoare până la valoarea maximă din șir, apoi sunt în ordine strict descrescătoare, după valoarea maximă din șir. Un șir strict crescător sau un șir strict descrescător nu se consideră șir **munte**. De exemplu, șirul de numere 2 8 5 3 sau șirul 1 2 6 4 3 2 este un șir **munte**, dar șirul 2 8 5 2 sau 2 8 5 5 sau 2 8 9 sau 2 8 8 5 nu este șir **munte**. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $2 < n < 200$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 10]$ , reprezentând notele obținute de  $n$  sportivi la un concurs. Programul afișează, pe ecran, pe linii distincte, notele concursului sortate crescător (separate printr-un spațiu), frecvențele fiecărei note obținute, de la 1 la 10 (separate printr-un spațiu), apoi mesajul **da**, dacă șirul, reprezentând frecvența notelor, este un șir **munte**, sau mesajul **nu**, în caz contrar.

**Exemplu:** dacă se citesc, în această ordine, separate prin **enter**, respectiv prin câte un spațiu, valorile

25

6 5 6 5 5 5 5 6 3 4 6 9 2 3 4 4 4 8 8 7 7 6 6 6 7

se va afișa

2 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 6 7 7 7 8 8 9

0 1 2 4 5 7 3 2 1 0

**da**

deoarece frecvența notelor de la 1 la 10 este șir **munte**

iar dacă se citesc, respectând același format de citire a datelor, valorile

10

6 5 6 5 10 6 3 3 9 9

se va afișa

3 3 5 5 6 6 6 9 9 10

0 0 2 0 2 3 0 0 2 1

**nu**

deoarece frecvența notelor nu este șir **munte**

### Limbajul Python

```

n=int(input())
note=list(map(int, input().split()))
rez=True
v=[0]*11
note=sorted(note)
print(*note)
for i in range(n):
 v[note[i]]+=1
print(*v)
pozmax=1
for i in range(11):
 if v[i]>v[pozmax]: pozmax=i

if pozmax==1 or pozmax==10:
 rez=False
else:
 for i in range(1,pozmax):

```

```

 if v[i]>=v[i+1]:
 rez=False
 for i in range (pozmax,10):
 if v[i]<=v[i+1]:
 rez=False

if rez: print("da")
else: print("nu")

```

### Limbajul C/C++

```

#include <iostream>
#include <algorithm>
using namespace std;
int main()
{
 int n,note[200],v[11]={0},i,j,pozmax;
 bool rez=true;
 cin>>n;
 for(i=0;i<n;i++)
 {cin>>note[i];
 v[note[i]]++;}
 sort(note,note+n);
 for(i=0;i<n;i++)
 cout<<note[i]<<' ';
 cout<<'\\n';
 for(i=1;i<11;i++)
 cout<<v[i]<<' ';
 cout<<'\\n';
 pozmax=1;
 for(i=2;i<=10;i++)
 if(v[i]>v[pozmax]) pozmax=i;
 if(pozmax==1 || pozmax==10)
 rez=false;
 else
 {
 for(i=1;i<pozmax;i++)
 if(v[i]>=v[i+1]) rez=false;
 for(i=pozmax;i<10;i++)
 if(v[i]<=v[i+1])
 rez=false;
 }
 if (rez) cout<<"da";
 else cout<<"nu";
 return 0;
}

```

33. Un număr se numește număr **nou** dacă cifele sale sunt ordonate strict descrescător și conține numai cifre consecutive. De exemplu, **654** este un număr **nou**, dar **643** sau **234** nu este un număr **nou**. Scrieți un program **Python/C++** care citește, de la tastatură, un număr natural **n**, având cel mult 300 de cifre. Programul afișează, pe ecran, numărul minim de numere **noi** care se pot forma, utilizând toate cifrele numărului **n**, ca în exemplul dat. Un număr cu o cifră se consideră număr **nou**.

De exemplu, pentru numărul **n=206124556**, programul va afișa **4** deoarece, cu toate cifrele lui **n**, cel mai scurt șir de numere **noi** care utilizează toate cifrele din **n** este: **2, 65, 210, 654**, iar acesta are lungimea **4**, iar pentru numărul **n=202**, programul va afișa **3** deoarece, cu toate cifrele lui **n**, cel mai scurt șir de numere **noi** este: **0, 2, 2** iar acesta are lungimea **3**.

### Limbajul Python

```
n = input("n=")
v = [0] * 11
for i in n:
 v[int(i)] += 1
rez=0
for i in range(9,-1,-1):
 if v[i+1]-v[i]<0:
 rez=rez+v[i]-v[i+1]
print(rez)
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char n[301];
 int v[11]={0},i,rez=0;
 cin.get(n,301);
 for(i=0;n[i];i++)
 v[n[i]-'0']++;
 for(i=9;i>=0;i--)
 if(v[i+1]-v[i]<0)
 rez=rez+v[i]-v[i+1];
 cout<<rez;
 return 0;
}
```

34. Se consideră un text cu cel mult 300 de caractere, format din cel puțin două cuvinte, în care cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program **Python/C++** care citește de la tastatură un text de tipul menționat mai sus și construiește în memorie un nou text astfel încât, din textul inițial, să se elimine toate cuvintele care apar de cel puțin două ori. Noul text are cuvintele în ordinea inversă față de ordinea inițială, separate tot printr-un singur spațiu. Programul afișează pe ecran noul text, ca în exemplu. Dacă toate cuvintele din textul inițial apar de cel puțin două ori, se va afișa pe ecran mesajul **vid**.

De exemplu, pentru textul citit: **a fi sau a nu fi**

se afișează pe ecran **nu sau**

iar pentru textul

**da nu nu da da**

se afișează pe ecran mesajul **vid**

### Limbajul Python

```
t = input(" Introduceți textul: ")
s = t.split(' ')
cs=s
nou = []
ok=False
for cuv in s:
 nr=0
 c=cuv
```

```

 for c in cs:
 if c==cuv: nr+=1
 if nr==1:
 nou.insert(0,cuv)
 ok=True
 if ok: print(*nou)
 else: print("vid")

```

## Limbajul C/C++

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char s[301], *cuv, cs[301], nou[301]="";
 cout<<"Introduceti textul: ";
 cin.get(s,301); strcpy(cs,s);
 cuv=strtok(s," ");
 while(cuv)
 {
 char copie[301],*p;
 strcpy(copie,cs);
 int nr=0;
 p=strstr(copie,cuv);
 while(p)
 {
 nr++;
 p=strstr(p+1,cuv);
 }
 if(nr==1)
 {
 char a[301];
 strcpy(a,nou);
 strcpy(nou,cuv);
 strcat(nou," ");
 strcat(nou,a);}
 cuv=strtok(0," ");
 }
 if(strlen(nou)) cout<<nou;
 else cout<<"vid";
 return 0;
}

```

35. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $1 \leq n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul afișează pe ecran, ca în exemplu, pe linii separate:
- Numărul de elemente pare și numărul de elemente impare din listă
  - Media aritmetică a tuturor valorilor pare
  - Elementul maxim aflat pe o poziție impară (poziții indexate de la 0)
  - Suma elementelor aflate pe poziții multipli de 3 (0, 3, 6, ...)

**De exemplu**, dacă se citesc numerele 5 12 7 9 4 2 11 18, programul afișează:

- Pare: 4, Impare: 3
- Media valorilor pare: 9.0
- Maxim pe poziții impare: 11
- Suma elementelor pe poziții multipli de 3: 34

## Limbajul Python

```
n = int(input("n = "))
while n < 1 or n >= 100:
 n = int(input("n : "))

L = []
print(f"Introduceți n:")
for i in range(n):
 x = int(input(f"Elementul {i + 1}: "))
 L.append(x)

a) Număr de pare și impare
nr_pare = nr_impere = 0
suma_pare = 0
for x in L:
 if x % 2 == 0:
 nr_pare += 1
 suma_pare += x
 else:
 nr_impere += 1

b) Media numerelor pare
media_pare = suma_pare / nr_pare if nr_pare != 0 else 0

c) Maxim pe poziții impare (indexuri: 1, 3, 5...)
max_impere = None
for i in range(1, n, 2):
 if max_impere is None or L[i] > max_impere:
 max_impere = L[i]

d) Suma pe poziții multipli de 3
suma_mult_3 = 0
for i in range(0, n, 3):
 suma_mult_3 += L[i]
print(f"a) Pare: {nr_pare}, Impare: {nr_impere}")
print(f"b) Media valorilor pare: {media_pare:.2f}")
print(f"c) Maxim pe poziții impare: {max_impere}")
print(f"d) Suma pe poziții multipli de 3: {suma_mult_3}")
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, L[100];
 cout << "n = "; cin >> n;
 for (int i = 0; i < n; i++) cin >> L[i];
 int nr_pare = 0, nr_impere = 0, suma_pare = 0;
 int max_impere = -1; // presupunem toate valorile > 0
 int suma_mult_3 = 0;
 for (int i = 0; i < n; i++) {
 if (L[i] % 2 == 0) { nr_pare++; suma_pare += L[i]; }
 else nr_impere++;
 if (i % 2 == 1 && L[i] > max_impere) max_impere = L[i];
 if (i % 3 == 0) suma_mult_3 += L[i];
 }
}
```

```

cout << "a) Pare: " << nr_pare << ", Impare: " << nr_impere;
if (nr_pare)
 cout << "b) Media valorilor pare: "<<(float)suma_pare/nr_pare;
else
 cout << "b) Nu exista valori pare\n";
 cout << "c) Maxim pe pozitii impare: " << max_impere;
 cout << "d) Suma pe pozitii multipli de 3: " << suma_mult_3;
return 0; }

```

36. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 1000]$ . Programul afișează lista inițială, apoi modifică lista eliminând toate numerele formate din trei cifre și afișează, pe ecran, lista modificată. Dacă lista rezultată este vidă, programul va afișa mesajul: **toate valorile au fost de 3 cifre**.

**De exemplu** dacă se citesc, în această ordine, separate prin enter, valorile:

666 213 421 7 800 92 101

se afișează

Lista initiala: 666 213 421 7 800 92 101

Lista finala: 7 92

### Limbajul Python

```

n = int(input("n = "))
L=[]
for i in range(n):
 x = int(input(f"Elementul {i+1}: "))
 L.append(x)
print("Lista inițială:", *L)
L_filtrat = [x for x in L if (x<100 or x>999)]
if L_filtrat==[]: print("toate valorile au fost de 3 cifre")
else: print("Lista finala: ", *L_filtrat)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main() {
 int n, x;
 int L[100]; // lista inițială
 int L2[100]; // listă pentru valorile impare
 int k = 0; // număr de valori impare
 cout << "n = "; cin >> n;
 for (int i = 0; i < n; i++) {
 cin >> x; L[i] = x;
 if (x<100 || x>999) L2[k++] = x; }
 cout << "Lista initiala: ";
 for (int i = 0; i < n; i++) cout << L[i] << " ";
 cout << endl;
 if (k == 0) cout << "toate valorile au fost de 3 cifre\n";
}

```

```

else { cout << "Lista finala: ";
 for (int i = 0; i < k; i++)cout << L2[i] << " ";
 cout << endl; }
return 0; }

```

37. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește două liste **L1** și **L2** cu  $n$  elemente, numere naturale citite de la tastatură. Numerele citite sunt naturale din intervalul  $[1, 100]$ . Programul afișează cele două liste, ca în exemplu, apoi construiește și afișează:

- o nouă listă **L3** formată din elementele **minime** aflate pe poziții omoloage:  
 $L3[i] = \min(L1[i], L2[i])$  ;
- O listă **L4** care să conțină valoarea 1 pe pozițiile în care elementele din L1 și L2 sunt **egale**, și 0 în rest;

**De exemplu** dacă se citesc, în această ordine, separate prin enter, valorile  $n=5$  L1 [10, 20, 30, 40, 50] și L2=[5, 20, 25, 60, 50], programul va afișa:

L1: [10, 20, 30, 40, 50]

L2: [5, 20, 25, 60, 50]

L3(minime): [5, 20, 25, 40, 50]

L4(egale): [0, 1, 0, 0, 1]

### Limbajul Python

```

n = int(input("n = "))
L1 = []
L2 = []
print("Introduceți elementele pentru lista L1:")
for i in range(n):
 x = int(input(f"L1[{i+1}] = "))
 L1.append(x)
print("Introduceți elementele pentru lista L2:")
for i in range(n):
 x = int(input(f"L2[{i+1}] = "))
 L2.append(x)
L3 = []
L4 = []
for i in range(n):
 L3.append(min(L1[i], L2[i]))
 L4.append(1 if L1[i] == L2[i] else 0)
print("L1:", ' '.join(map(str, L1)))
print("L2:", ' '.join(map(str, L2)))
print("L3 (minime):", ' '.join(map(str, L3)))
print("L4 (egale):", ' '.join(map(str, L4)))

```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, L1[100], L2[100], L3[100], L4[100];
 cout << "n = ";
 cin >> n;
 cout << "Introduceti elementele pentru L1:\n";
 for (int i = 0; i < n; i++) cin >> L1[i];
 cout << "Introduceti elementele pentru L2:\n";
 for (int i = 0; i < n; i++) cin >> L2[i];
 for (int i = 0; i < n; i++) {
 L3[i] = (L1[i] < L2[i]) ? L1[i] : L2[i];
 L4[i] = (L1[i] == L2[i]) ? 1 : 0;
 }
 cout << "L1: ";
 for (int i = 0; i < n; i++) cout << L1[i] << " ";
 cout << "\nL2: ";
 for (int i = 0; i < n; i++) cout << L2[i] << " ";
 cout << "\nL3 (minime): ";
 for (int i = 0; i < n; i++) cout << L3[i] << " ";
 cout << "\nL4 (egale): ";
 for (int i = 0; i < n; i++) cout << L4[i] << " ";
 cout << endl;
 return 0;
}
```

38. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și un șir de  $n$  numere întregi din intervalul  $[-1000, 1000]$ . Programul construiește o listă care să conțină numerele din șirul citit, în ordinea citirii, apoi modifică lista eliminând toate elementele negative. Programul afișează pe ecran lista inițială și lista modificată, ca în exemplu.

**De exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile

5 -7 10 -3 4 -1

programul afișează:

Li: [-7, 10, -3, 4, -1]

Lf: [0, 10, 0, 4, 0]

## Limbajul Python

```
n = int(input("n = "))
L = []
for i in range(n):
 x = int(input(f"Elementul {i+1}: "))
 L.append(x)
print("Lista inițială:")
```

```

print(' '.join(map(str, L)))
for i in range(n):
 if L[i] < 0:
 L[i] = 0
print("Lista după înlocuirea valorilor negative cu 0:")
print(' '.join(map(str, L)))

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main() {
 int n, L[100];
 cout << "n = "; cin >> n;
 for (int i = 0; i < n; i++) cin >> L[i];
 cout << "Li: ";
 for (int i = 0; i < n; i++) cout << L[i] << " ";
 for (int i = 0; i < n; i++) if (L[i] < 0) L[i] = 0;
 cout << "\nLf: ";
 for (int i = 0; i < n; i++) cout << L[i] << " ";
 cout << "\n";
 return 0;
}

```

39. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul  $n$  ( $n < 100$ ) și care construiește o listă cu  $n$  numere citite de la tastatură. Numerele citite sunt naturale din intervalul  $[0, 1000]$ . Programul modifică lista inserând după fiecare număr cifra sa maximă, apoi afișează lista inițială și lista modificată.

**De exemplu:** dacă pentru  $n$  se citește 4, iar pentru listă se citesc valorile: 12 40 5 103 se afișează

Li: [12, 40, 5, 103]

Lm: [12, 2, 40, 4, 5, 5, 103, 3]

### Limbajul Python

```

def cifra_maxima(x):
 return max(int(c) for c in str(x))

n = int(input("Introduceți n (<100): "))
Li = []
print("Introduceți cele", n, "numere (între 0 și 1000):")
for _ in range(n):
 x = int(input())
 Li.append(x)
Lm = []

```

```

for num in Li:
 Lm.append(num)
 Lm.append(cifra_maxima(num))
print("Li:", Li)
print("Lm:", Lm)

```

## Limbajul C/C++

```

#include <iostream>
using namespace std;

int suma_cifre(int x) {
 int s = 0;
 while (x > 0)
 {s += x % 10;
 x /= 10;
 }
 return s;
}

int main() {
 int n, x, int Li[100], Lm[200], j = 0, i;
 cout << "n = ";
 cin >> n;
 for (i = 0; i < n; i++) cin >> Li[i];
 cout << "Li: ";
 for (int i = 0; i < n; i++)
 cout << Li[i] << " "; // Afișăm lista inițială
 cout << "\n";
 for (i = 0; i < n; i++) {
 Lm[j++] = Li[i]; Lm[j++] = suma_cifre(Li[i]);
 }
 cout << "\nLi: ";
 for (int i = 0; i < j; i++)
 cout << Lm << " ";
 return 0;
}

```

40. Scrieți un program **Python/C++** care citește de la tastatură un număr natural nenul par  $n$  ( $n < 100$ ) și  $n$  numere întregi din intervalul  $[-1000, 1000]$ . Programul construiește o listă **L1** care să conțină cele  $n$  numere, în ordinea citirii, apoi construiește o nouă listă **L2** cu elementele primei liste în ordinea **L1[0]**, **L1[n-1]**, **L1[1]**, **L1[n-2]**, **L1[2]**, **L1[n-3]**, ... etc. Programul afișează pe ecran cele două liste create, pe linii separate, ca în exemplu.

**De exemplu:** dacă se citesc, în această ordine, separate prin enter, valorile

6 1 2 3 4 5 6

programul afișează:

**Li:** 1 2 3 4 5 6

**Lf:** 1 6 2 5 3 4

## Limbajul Python

```
n = int(input("n= "))
L1 = []
for i in range(n):
 print
 x = int(input("L1["+str(i)+"]="))
 L1.append(x)
L2 = []
for i in range(n // 2):
 L2.append(L1[i])
 L2.append(L1[n - 1 - i])
print("L1:", *L1)
print("L2:", *L2)
```

## Limbajul C/C++

```
#include <iostream>
using namespace std;
int main() {
 int n, L1[100], L2[100], i, k=0;
 cout << "n= "; cin >> n;
 for (int i = 0; i < n; i++)
 { cout<<"L1["<<i<<"="; cin>>L1[i]; }
 for (int i = 0; i < n / 2; i++) {
 L2[k++] = L1[i];
 L2[k++] = L1[n - 1 - i]; }
 cout << "L1:";
 for (int i = 0; i < n; i++) cout << " " << L1[i];
 cout << endl << "L2:";
 for (int i = 0; i < n; i++) cout << " " << L2[i];
 cout << endl;
 return 0;
}
```

41. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $5 < n < 2025$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 10000]$ , reprezentând punctajele celor  $n$  elevi participanți la concursul de proiecte științifice. La acest concurs au fost acordate de către comisie premii speciale pentru elevii care au **punctaj perfect**. Un punctaj este perfect dacă numărul format din prima cifră și cifra zecilor este un număr pătrat perfect și numărul este cu exact patru cifre. Afișați pe primul rând, punctajele primilor 5 **elevi** premiați la concurs, cu un spațiu între punctaje, respectiv numărul premiilor speciale acordate de către comisie pe rândul următor.

**Exemplu:** dacă se citesc, în această ordine, separate prin **enter**, respectiv prin câte un spațiu, valorile

10

2985 1016 9031 3006 7123 7894 8031 685 8881 644

Se va afișa

9031 8881 8031 7894 7123

5

## Limbajul Python

```
import math
n=int(input())
v=list(map(int, input().split()))
nr=0
for x in v:
 if x>999 and x<=9999:
 y=(x//1000)*10+x%10
 rady = int(math.sqrt(y))
 if rady*rady==y:
 nr=nr+1
v.sort(reverse=True)
print(*v[:5])

#v[:5] - selectează primele 5 elemente (sau mai puține dacă lista are
#mai puțin de 5)
#* - „despachetează” lista astfel încât valorile să fie afișate
#spațiate, nu ca o listă între paranteze.

print(nr)
```

## Limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
 int n,v[2025],i,j,nr=0,y,rady;
 cin>>n;
 for(i=0;i<n;i++)
 cin>>v[i];
 for(i=0;i<n-1;i++)
 for(j=i+1;j<n-1;j++)
 if(v[i]<v[j])
 {
 int aux=v[i];
 v[i]=v[j];
 v[j]=aux;
 }
 for(i=0;i<5;i++)
 cout<<v[i]<<" ";
 cout<<endl;
 for(i=0;i<n;i++)
 {
 if(v[i]>999 && v[i]<10000)
 {
 y=(v[i]/1000)*10+v[i]%10;
 rady = int(sqrt(y));
 if (rady*rady==y) nr=nr+1;
 }
 }
 cout<<nr;
 return 0;
}
```

42. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $5 < n < 2025$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 10000]$ , reprezentând punctajele celor  $n$  sportivi participanți la jocurile olimpice de vară. La această olimpiadă au fost acordate de către organizatori premii surpriză pentru sportivii care au **primcod**. Un sportiv are **primcod** dacă suma cifrelor punctajului obținut este un număr prim. Afișați pe primul rând, punctajele obținute de sportivi, în ordine crescătoare a punctajelor, cu un spațiu între punctaje, respectiv numărul premiilor surpriză acordate de către comisie pe rândul următor.

**Exemplu:** dacă se citesc, în această ordine, separate prin **enter**, respectiv prin câte un spațiu, valorile

```
10
2985 1016 9031 3006 7123 7894 8031 685 8881 644
Se va afișa
644 685 1016 2985 3006 7123 7894 8031 8881 9031
3
```

### Limbajul Python

```
def este_prim(x):
 if x < 2:
 return False
 for d in range(2, int(x ** 0.5) + 1):
 if x % d == 0:
 return False
 return True

n = int(input(""))
v = list(map(int, input().split()))

Sortare crescatoare
v.sort()

Afișare lista sortata fara paranteze
print(*v)

Calculez suma cifrelor fiecarui element al listei
nr = 0
for x in v:
 s = sum(int(cifra) for cifra in str(abs(x)))
 # verific daca suma cifrelor este nr prim
 if este_prim(s):
 nr += 1
print(nr)
```

### Limbajul C/C++

```
#include <iostream>
using namespace std;
int main()
{
 int n, v[2025], i, j, nr=0, aux, c, s, k;
 cin >> n;
 for(i=0; i<n; i++)
 cin >> v[i];
 for(i=0; i<n-1; i++)
 for(j=i+1; j<n; j++)
 if(v[i]>v[j])
 {
 aux=v[i];
 v[i]=v[j];
 v[j]=aux;
 }
}
```

```

 for(i=0;i<n;i++)
 cout<<v[i]<<" ";
 cout<<endl;
 for(i=0;i<n;i++)
 {
 c=v[i];
 s=0;
 while(c)
 {
 s=s+c%10; c=c/10;
 }
 k=1;
 for(j=2;j*j<=s&& k==1;j++)
 if(s%j==0)
 k=0;
 if (k) nr=nr+1;
 }
 cout<<nr;
 return 0;
}

```

43. Scrieți un program **Python/C++** care citește de la tastatură un număr natural  $n$  ( $5 < n < 2025$ ) și un șir de  $n$  numere naturale din intervalul  $[1, 10000]$ , reprezentând punctajele celor  $n$  șahiști participanți la campionatul național de șah. La acest campionat au fost eliminați din competiție ultimii 5 șahiști, în funcție de obținut obținut. Afișați pe primul rând, punctajele obținute de sportivi, în ordine crescătoare a punctajelor, cu un spațiu între punctaje. Pe al doilea rând afișați punctajele șahiștilor rămași după eliminare, cu spații între punctaje. Pe rândul a treilea afișați rezultatele șahiștilor astfel punctajele pare ordonate crescător, iar punctajele impare descrescător, ca în exemplu.

**Exemplu:** dacă se citesc, în această ordine, separate prin **enter**, respectiv prin câte un spațiu, valorile:

12

2985 1016 9031 3006 7123 7894 8031 685 8881 644 7444 8888

Se va afișa:

644 685 1016 2985 3006 7123 7444 7894 8031 8881 8888 9031

7123 7444 7894 8031 8881 8888 9031

9031 7444 7894 8881 8031 8888 7123

### Limbajul Python

```

n = int(input())
v = list(map(int, input().split()))

Sortare manuală crescătoare (bubble sort)
for i in range(n - 1):
 for j in range(i + 1, n):
 if v[i] > v[j]:
 v[i], v[j] = v[j], v[i]

Afișare vector sortat
print(*v)

Elimin primele 5 elemente
if n > 5: # trebuie să avem cel puțin 5 elemente

```

```

 for i in range(4, n):
 if i + 1 < len(v):
 v[i - 4] = v[i + 1]
 n -= 5
 v = v[:n] # redimensionăm lista la noua lungime
else:
 print("Sunt prea puține elemente pentru a elimina 5 punctaje!")
 v = v[:n] # în caz că nu putem elimina

Afișare vector după eliminare
print(*v)

for i in range(n - 1):
 for j in range(i + 1, n):
 if v[i] > v[j] and v[i]%2==0 and v[j] % 2 == 0:
 v[i], v[j] = v[j], v[i]

for i in range(n - 1):
 for j in range(i + 1, n):
 if v[i] < v[j] and v[i]%2==1 and v[j] % 2 == 1:
 v[i], v[j] = v[j], v[i]

Afișare vector final
print(*v)

```

### Limbajul C/C++

```

#include <iostream>
using namespace std;
int main()
{
 int n,v[2025],i,j,nr=0,aux;
 cin>>n;
 for(i=0;i<n;i++)
 cin>>v[i];
 for(i=0;i<n-1;i++)
 for(j=i+1;j<n;j++)
 if(v[i]>v[j])
 {
 aux=v[i]; v[i]=v[j]; v[j]=aux;
 }
 for(i=0;i<n;i++) cout<<v[i]<<" ";
 cout<<endl;
 for(i=4;i<n;i++) v[i-4]=v[i+1];
 n=n-5;
 for(i=0;i<n;i++) cout<<v[i]<<' ';
 cout<<endl;
 for(i=0;i<n-1;i++)
 for(j=i+1;j<n;j++)
 if(v[i]>v[j] && v[i]%2==0 &&v[j]%2==0)
 {
 aux=v[i]; v[i]=v[j]; v[j]=aux;
 }
 for(i=0;i<n-1;i++)
 for(j=i+1;j<n;j++)
 if(v[i]<v[j] && v[i]%2==1 &&v[j]%2==1)
 {
 aux=v[i]; v[i]=v[j]; v[j]=aux;
 }
 for(i=0;i<n;i++) cout<<v[i]<<" ";
 return 0;
}

```

43. Scrieți un program Python/C++ care citește de la tastatură un număr natural  $n$  și cele  $n$  cuvinte care reprezintă denumiri de fructe. Un cuvânt este format din cel mult 31 de caractere, doar litere mici ale alfabetului englez. Programul afișează pe ecran, pe linii separate, denumirile fructelor care încep cu litera  $c$ .

**De exemplu**, pentru  $n=5$  și cuvintele:

mere  
pere  
prune  
caise

castana programul va afișa:

- În Python  
['mere', 'pere', 'prune', 'caise', 'castana']  
caise  
castana
- În C++  
mere pere prune caise castana  
caise  
castana

### Limbajul Python

```
liste creez lista cu fructe
n=int(input("dati n= "))
fructe=[]
for i in range(0,n):
 sir=input("fructul =")
 fructe.append(sir)
print(fructe)
#afisez cuvintele care incep cu c
for i in range(0,n):
 if fructe[i][0]=='c': print(fructe[i])
```

### Limbajul C/C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char s[31],c,fructe[31][31];
 int n,i;
 cin>>n;
 for(i=1; i<=n;i++)
 {
 cin>>s;
 strcpy(fructe[i],s); cout<<fructe[i]<<" ";
 cout<<endl;
 for(i=1; i<=n;i++)
 if(fructe[i][0]=='c') cout<<fructe[i]<<" \n";
 return 0;
 }
}
```

## VII. Fişiere Text

1. Fişierul **date.in** conţine, în ordine crescătoare, cel puţin două şi cel mult 1000 de numere naturale. Numerele sunt separate prin câte un spaţiu şi au cel mult 6 cifre fiecare. Cel puţin un număr din fişier este par şi apare o singură dată.

Scrieţi un program **Python/C++** care citeşte toate numerele din fişier şi, printr-un algoritm eficient din punctul de vedere al timpului de executare şi al memoriei utilizate, determină şi afişează pe ecran, în ordine strict crescătoare, separate prin câte un spaţiu, numai numerele pare care apar o singură dată în fişier.

**Exemplu:** dacă fişierul conţine numerele:

1 1 2 2 2 6 7 8 10 10 10 12 12 21

se afişează, în această ordine, numerele 6 8.

### Limbajul Python

```
with open("date.in", "r") as fin:
 numere = map(int, fin.read().split())
 ult=None
 cnt=0
 for numar in numere:
 if numar==ult:
 cnt += 1
 else:
 if ult is not None and ult % 2 == 0 and cnt == 1:
 print(ult, end=" ")
 ult=numar
 cnt = 1
 if ult is not None and ult % 2 == 0 and cnt == 1:
 print(ult)
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;

ifstream fin("date.in");
int main()
{
 int crt,ult=-1,cnt=0;
 while(fin>>crt)
 {
 if(crt==ult) ++cnt;
 else
 {
 if(ult%2==0 && cnt==1)
 cout<<ult<<' ';
 ult=crt;
 cnt=1;
 }
 }
 if(ult%2==0 && cnt==1)
 cout<<ult<<' ';
 fin.close();
 return 0;
}
```

2. Fișierul `date.in` conține pe prima linie un număr natural  $n$  ( $3 < n < 1000$ ), iar pe următoarea linie, un șir de  $n$  numere naturale distincte, de cel mult șase cifre fiecare. Numerele din șir sunt separate prin câte un spațiu și cel puțin trei dintre ele sunt impare. Scrieți un program **Python/C++** care citește toate numerele din fișier și, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, determină și afișează pe ecran cele mai mari trei numere din șir care sunt impare. Numerele determinate sunt afișate în ordine descrescătoare, separate prin câte un spațiu.

**Exemplu:** dacă fișierul conține numerele:

11

23 12 32 31 31 56 43 79 345 13 17

se afișează, în această ordine, numerele 345 79 43.

### Limbajul Python

```
with open("date.in", "r") as f:
 n=int(f.readline())
 linie=f.readline().split()
 max1 = max2 = max3 = -1
 for x in linie:
 nr=int(x)
 if nr%2==1:
 if nr>max1:
 max3=max2
 max2=max1
 max1=nr
 elif nr>max2:
 max3=max2
 max2=nr
 elif nr>max3: max3=nr
 print(max1,max2,max3)
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
 ifstream fin("date.in");
 int n,x,max1=-1,max2=-1,max3=-1;
 fin>>n;
 for(int i=0; i<n; ++i)
 {
 fin>>x;
 if(x%2==1)
 if(x>max1)
 { max3=max2; max2=max1; max1=x;}
 else if(x>max2)
 { max3=max2; max2=x; }
 else if(x>max3) max3=x;
 }
 cout<<max1<<" "<<max2<<" "<<max3;
 return 0;
}
```

3. Scrieți un program **Python/C++** care citește din fișierul `date.in` două numere naturale  $s_1$  și  $s_2$  ( $0 < s_1 \leq 18$ ,  $0 \leq s_2 \leq 18$ ) și care afișează pe ecran cel de-al treilea numărului natural cu exact 4 cifre, pentru care suma primelor două cifre este egală cu  $s_1$ , iar suma ultimelor două cifre este egală cu  $s_2$ . Pentru determinarea numărului se utilizează

un algoritm eficient din punct de vedere al timpului de executare. Dacă nu există acest număr, pe ecran se afișează mesajul **imposibil**.

**Exemplu:** dacă  $s_1=5$ , iar  $s_2=7$ , atunci se afișează **1425** (primele trei numere sunt **1407 1416 1425**).

### Limbajul Python

```
with open("date.in","r") as f:
 s1,s2=map(int,f.readline().split())
 cnt=0
 for a in range(1,10):
 if a>s1:break
 b=s1-a
 if 0<=b<=9:
 for c in range(0,10):
 if c>s2: break
 d=s2-c
 if 0<=d<=9:
 n=a*1000+b*100+c*10+d
 cnt+=1
 if cnt==3:
 print(n)
 exit()
print("imposibil")
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
 ifstream f("date.in");
 int s1,s2,cnt=0;
 f>>s1>>s2;
 for(int a=1; a<=9&&a<=s1; ++a)
 {
 int b=s1-a;
 if(b>=0&&b<=9)
 for(int c=0; c<=9&&c<=s2; ++c)
 {
 int d=s2-c;
 if(d>=0&&d<=9)
 {
 int n=a*1000+b*100+c*10+d;
 ++cnt;
 if(cnt==3)
 {
 cout<<n<<endl; return 0; }
 }
 }
 }
 cout<<"imposibil";
 return 0;
}
```

4. Fișierul **date.in** conține un șir de cel puțin 11 și cel mult o mie de numere naturale, separate prin câte un spațiu. Fiecare număr are cel puțin două și cel mult șase cifre. Primul termen al șirului are numărul de ordine 1, al doilea are numărul de ordine 2 etc. Scrieți un program **Python/C++** care citește șirul din fișier și, utilizând un algoritm eficient din punctul de vedere al timpului de executare, determină și afișează pe ecran numărul de ordine al unui termen al șirului care este precedat în fișier de un număr maxim de valori care au cifra unităților egală cu a sa. Dacă sunt mai mulți termeni cu această proprietate, se afișează numărul de ordine doar al unuia dintre ei.

**Exemplu:** dacă fișierul `date.in` conține numerele 21 63 256 81 193 91 23 43 121 41 86 pe ecran se afișează 10 (numărul de ordine al termenului 41).

### Limbajul Python

```
with open("/date.in","r") as f:
 freqv=[0]*10
 maxim=-1
 poz=-1
 i=0
 for linie in f:
 for val in linie.split():
 x=int(val)
 uc=x%10
 i+=1
 if freqv[uc]>maxim:
 maxim=freqv[uc]
 poz=i
 freqv[uc]+=1
print(poz)
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
 ifstream fin("date.in");
 int x,freqv[10]= {0},maxim=-1,poz=-1,i=0;
 while(fin>>x)
 {
 int uc=x%10;
 ++i;
 if(freqv[uc]>maxim)
 {
 maxim=freqv[uc];
 poz=i;
 }
 ++freqv[uc];
 }
 cout<<poz;
 return 0;
}
```

5. Fiind date două numere `a` și `b`, îl numim pe `a` sufix al lui `b` dacă `a` este egal cu `b` sau dacă `b` se poate obține din `a` prin alipirea la stânga a unor noi cifre. Exemplu: 17 este sufix al lui 17, iar 45 este sufix al lui 78145.

Fișierul `date.in` conține pe prima linie un număr natural `x`, cu exact două cifre, iar pe a doua linie un șir de cel puțin două și cel mult 1000 de numere naturale cu cel mult șase cifre. Numerele din șir sunt separate prin câte un spațiu.

Scrieți un program **Python/C++** care să afișeze pe ecran al doilea termen al șirului care are ca sufix numărul `x`. Dacă în șir nu există cel puțin două asemenea valori, pe ecran se afișează mesajul **nu exista**. Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**Exemplu:** dacă fișierul `date.in` are conținutul

45

3445 89312 1245 12 67120 31345 123457 atunci pe ecran se afișează 1245.

## Limbajul Python

```
with open("date.in","r") as f:
 x=int(f.readline())
 cnt=0
 for val in f.readline().split():
 n=int(val)
 if n%100==x : cnt+=1
 if cnt==2:
 print(n)
 exit()
print("nu exista")
```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
 ifstream fin("date.in");
 int x,cnt=0,n;
 fin>>x;
 while(fin>>n)
 if(n%100==x)
 {
 ++cnt;
 if(cnt==2)
 {
 cout<<n;
 return 0;
 }
 }
 cout<<"nu exista";
 return 0;
}
```

6. Numim secvență egală a unui șir de numere naturale un subșir al acestuia, format din termeni pari cu aceeași valoare, aflați pe poziții consecutive în șirul dat. Lungimea secvenței este egală cu numărul de termeni ai acesteia.

Fișierul `date.in` conține un șir de cel puțin două și cel mult o mie de numere naturale din intervalul  $[0, 10^6]$ . Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni egali pe poziții consecutive.

Scrieți un program **Python/C++** care să determine o secvență egală de lungime maximă în șirul aflat în fișier și să afișeze pe ecran lungimea acestei secvențe și, pe o linie nouă, separați prin câte un spațiu, termenii acesteia. Dacă sunt mai multe astfel de secvențe, se afișează doar termenii primei dintre acestea, iar dacă nu există nicio asemenea secvență se afișează mesajul **lipsa pare**. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei necesare și al timpului de executare.

**Exemplu:** dacă fișierul `date.in` conține numerele

5 5 5 2 2 2 2 7 3 3 3 3 8 8 8 8 1 1 8 8 8 8

atunci pe ecran se afișează valorile

4

2 2 2 2

## Limbajul Python

```
with open("date.in") as f:
 val_crt=-1
 Lg_crt=0
 Val=-1
 LgMax=0
 gasit=False
 for val in f.read().split():
 x=int(val)
 if x%2==0:
 gasit=True
 if x==val_crt:
 Lg_crt+=1
 if Lg_crt>LgMax:
 LgMax=Lg_crt
 Val=val_crt
 else:
 val_crt=x
 Lg_crt=1
 else:
 val_crt=-1
 Lg_crt=0
 if gasit:
 print(LgMax)
 print(" ".join([str(Val)]*LgMax))
 else: print("lipsa pare")
```

## Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{ ifstream f("date.in");
 int x,val_crt=-1,Lg_crt=0,Val=-1, LgMax=0;
 bool gasit=false;
 while(f>>x)
 { if(x%2==0)
 { gasit=true;
 if(x==val_crt)
 {
 Lg_crt++;
 if(Lg_crt>LgMax) { LgMax=Lg_crt; Val=val_crt;}
 }
 else { val_crt=x; Lg_crt=1; }
 }
 else { val_crt=-1; Lg_crt=0; }
 }
 if(gasit) { cout<<LgMax<<"\n";
 for(int i=0; i<LgMax; ++i) cout<<Val<<" "; }
 else cout<<"lipsa pare";
 return 0;
}
```

7. Fișierul `date.in` conține numere naturale din intervalul  $[1, 10^4]$ : pe prima linie numărul  $n$ , pe a doua linie un șir de  $n$  numere ordonate strict descrescător, iar pe a treia linie două numere,  $x$  și  $y$  ( $x \leq y$ ). Numerele de pe aceeași linie sunt separate prin câte un spațiu.

Scrieți un program **Python/C++** care să afișeze pe ecran cel mai mare număr din șir care aparține intervalului  $[x, y]$ . Dacă nu există un astfel de număr, se afișează pe ecran mesajul `nu exista`. Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele

7

200 167 100 56 40 34 6

35 90

atunci pe ecran se afișează 56

### Limbajul Python

```
with open("date.in") as f:
 n=int(f.readline())
 v=list(map(int,f.readline().split()))
 x,y=map(int,f.readline().split())
 st,dr,poz=0,n-1,-1
 while st<=dr:
 m=(st+dr)//2
 if v[m]<=y:
 poz=m
 dr=m-1
 else: st=m+1
 if poz!=-1 and v[poz]>=x: print(v[poz])
 else: print("nu exista")
```

### Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
 ifstream f("date.in");
 int n,x,y,v[1000];
 f>>n;
 for(int i=0; i<n; ++i)f>>v[i];
 f>>x>>y;
 int st=0,dr=n-1,poz=-1;
 while(st<=dr)
 {
 int m=(st+dr)/2;
 if(v[m]<=y) { poz=m; dr=m-1; }
 else st=m+1;
 }
 if(poz!=-1&&v[poz]>=x) cout<<v[poz];
 else cout<<"nu exista";
 return 0;
}
```

8. Fișierul `date.in` conține numere naturale din intervalul  $[1, 10^4]$ : pe prima linie numărul  $n$ , pe a doua linie un șir de  $n$  numere ordonate strict descrescător, iar pe a treia linie două numere,  $x$  și  $y$  ( $x \leq y$ ). Numerele de pe aceeași linie sunt separate prin câte un spațiu.

Scrieți un program **Python/C++** care să afișeze pe ecran, în ordine crescătoare, toate numerele din șir care aparțin intervalului  $[x, y]$ . Dacă nu există un astfel de număr, se afișează pe ecran mesajul `nu exista`. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele

7

200 167 100 56 40 34 6

35 90

atunci pe ecran se afișează 40 56

### Limbajul Python

```
with open("date.in") as f:
 n=int(f.readline())
 v=list(map(int,f.readline().split()))
 x,y=map(int,f.readline().split())
 st,dr=0,n-1
 poz_st=-1
 while st<=dr:
 m=(st+dr)//2
 if v[m]<=y:
 poz_st=m
 dr=m-1
 else: st=m+1
 st,dr=0,n-1
 poz_dr=-1
 while st<=dr:
 m=(st+dr)//2
 if v[m]>=x:
 poz_dr=m
 st=m+1
 else: dr=m-1
 if poz_st==-1 or poz_dr==-1 or poz_st>poz_dr:
 print("nu exista")
 else:
 for i in range(poz_dr,poz_st-1,-1):
 print(v[i],end=" ")
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
 ifstream f("date.in");
 int n,x,y,v[1000];
 f>>n;
 for(int i=0; i<n; ++i) f>>v[i];
 f>>x>>y;
 int st=0,dr=n-1,poz_st=-1;
 while(st<=dr)
 {
 int m=(st+dr)/2;
 if(v[m]<=y) { poz_st=m; dr=m-1; }
 else st=m+1;
 }
 st=0;
```

```

dr=n-1;
int poz_dr=-1;
while(st<=dr)
{
 int m=(st+dr)/2;
 if(v[m]>=x)
 { poz_dr=m; st=m+1;}
 else dr=m-1;
}
if(poz_st==-1 || poz_dr==-1 || poz_st>poz_dr)
{ cout<<"nu exista"; return 0; }
for(int i=poz_dr; i>=poz_st; --i) cout<<v[i]<<" ";
return 0;
}

```

9. Fișierul **date.in** conține un șir de cel mult un milion de numere întregi din intervalul  $[-10^6, 10^6]$ , separate prin câte un spațiu. Scrieți un program Python/C++ care afișează pe ecran, separate printr-un spațiu, cel mai mare număr impar și ultimul număr par din fișier. Dacă nu există două astfel de numere, programul afișează pe ecran un mesaj corespunzător, astfel: **lipsă impar**, dacă în fișier sunt doar numere pare, **lipsa par** dacă în fișier sunt doar numere impare. Programul utilizează un algoritm eficient din punctul de vedere al timpului de executare și al memoriei necesare.

**Exemplu:** dacă fișierul **date.in** conține numerele:

20 -5 67 89 234 -891 -4 -345 73

se afișează pe ecran 89 -4

iar dacă fișierul conține numerele 9 -3 67 -93 -357

se afișează pe ecran mesajul **lipsa par**.

## Limbajul Python

```

with open("date.in", "r") as f:
 maxImpar = -10000001
 ultimPar = 10000001
 e_Impar = False
 e_Par = False
 for linie in f:
 for part in linie.split():
 x = int(part)
 if x % 2 != 0:
 if x > maxImpar:
 maxImpar = x
 e_Impar = True
 else:
 ultimPar = x
 e_Par = True
 if e_Impar and e_Par:
 print(f"{maxImpar} {ultimPar}")
 elif not e_Impar:
 print("lipsa impar")
 else:
 print("lipsa par")

```

## Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
 ifstream f("date.in");
 int x,maxImpar=-10000001,ultimPar=10000001;
 bool e_Impar=false;
 bool e_Par=false;
 while(f>>x)
 {
 if(x%2!=0)
 {
 if(x>maxImpar)
 maxImpar=x;
 e_Impar=true;
 }
 else
 {
 ultimPar=x;
 e_Par=true;
 }
 }
 if(e_Impar&&e_Par)
 cout<<maxImpar<<" "<<ultimPar;
 else if(!e_Impar)
 cout<<"lipsa impar";
 else
 cout<<"lipsa par";
 return 0;
}
```

10. Fișierul `date.in` conține un șir de cel mult  $10^6$  numere naturale din intervalul  $[1, 10^6]$ . Numerele din șir sunt ordonate strict crescător și sunt separate prin câte un spațiu. Scrieți un program **Python/C++** care să determine valorile naturale distincte din intervalul  $[pr, ult]$ , unde `pr` este primul număr din fișier iar `ult` este ultimul număr din fișier, care **nu** se găsesc în șirul menționat mai sus. Valorile determinate se scriu în fișierul `date.out` în ordine strict crescătoare, separate prin câte un spațiu. Dacă nu există astfel de valori, se scrie în fișier mesajul `complet`. Pentru determinarea valorilor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**Exemplu:** dacă fișierul `date.in` conține numerele: 3 4 6 8 11 18 22

în fișierul `date.out` sunt scrise valorile:

5 7 9 10 12 13 14 15 16 17 19 20 21

## Limbajul Python

```
fin=open("date.in","r")
fout=open("date.out","w")
linie=fin.readline()
gasit=False
it=iter(linie.split())
a=int(next(it))
```

```

for s in it:
 x=int(s)
 for i in range(a+1,x):
 fout.write(str(i)+" ")
 gasit=True
 a=x
if not gasit: fout.write("complet")
fin.close()
fout.close()

```

## Limbajul C/C++

```

#include<iostream>
#include<fstream>
using namespace std;
int main()
{
 ifstream fin("date.in");
 ofstream fout("date.out");
 int x,a;
 bool gasit=false;
 fin>>a;
 while(fin>>x)
 {
 for(int i=a+1; i<x; ++i)
 {
 fout<<i<<" ";
 gasit=true;
 }
 a=x;
 }
 if(!gasit)
 fout<<"complet";
 return 0;
}

```

11. Fişierele `date1.in` şi `date2.in` conţin câte un şir de  $n$  numere naturale din intervalul  $[1, 10^6]$ . Numerele din fişiere sunt separate prin câte un spaţiu. Scrieţi un program **Python/C++** care citeşte de la tastatură valoarea lui  $n$ , apoi scrie în fişierul `date.out` valorile naturale obţinute din însumarea valorilor aflate pe poziţii omoloage în fişierele `date1.in` şi `date2.in`. Valorile determinate se scriu separate prin câte un spaţiu. Pentru determinarea valorilor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei şi al timpului de executare.

**Exemplu:** pentru  $n=4$ , dacă fişierul `date1.in` conţine numerele: 3 6 8 10, iar fişierul `date2.in` conţine numerele: 9 11 1 15

În fişierul `date.out` sunt scrise valorile: 12 17 9 25

## Limbajul Python

```

Citeşte valoarea n de la tastatură: numărul de perechi de numere care
vor fi procesate
n=int(input())
se deschid fişierele "date1.in" şi "date2.in" în modul citire

```

```

f1=open("date1.in","r")
f2=open("date2.in","r")
fout=open("date.out","w") #se deschide fişierul "date.out" pentru scriere
i=0 # inițializare contorul
se repetă pentru fiecare pereche de numere (de la 0 la n-1)
while i<n:
 numar1="" # citire caracter cu caracter din f1 până
 se obține numărul
 while True:
 c=f1.read(1) # citește un caracter
 if not c: break # dacă s-a ajuns la sfârșitul fişierului
 if c.isspace(): # dacă e spațiu
 if numar1!="": break # dacă s-a format număr1
 else:numar1+=c # se adaugă cifra la număr1
 a=int(numar1) # se transformă şirul în număr întreg
 numar2="" # la fel pentru fişierul f2
 while True:
 c=f2.read(1)
 if not c: break
 if c.isspace():
 if numar2!="": break
 else: numar2+=c
 b=int(numar2)
 # se scrie suma celor două numere în fişierul fout, urmată de un spațiu
 fout.write(str(a+b)+" ")
 i+=1 # se trece la următoarea pereche
se închid toate fişierele
f1.close()
f2.close()
fout.close()

```

## Limbajul C/C++

```

#include<iostream> // pentru cin și cout
#include<fstream> // pentru ifstream și ofstream
using namespace std;
int main()
{
 int n,a,b; // n - numărul de perechi de numere de citit, a și b - valorile citite
 cin>>n; // se citește de la tastatură valoarea lui n
 ifstream f1("date1.in"); // se deschide fişierul date1.in pentru citire
 ifstream f2("date2.in"); // se deschide fişierul date2.in pentru citire
 ofstream fout("date.out"); // se deschide fişierul date.out pentru scriere
 for(int i=0;i<n;++i) // pentru fiecare poziție de la 0 la n-1
 {
 f1>>a; // se citește un număr din primul fişier
 f2>>b; // se citește un număr de pe aceeași poziție din al doilea fişier
 fout<<a+b<<" "; // se scrie suma celor două numere în fişierul de ieşire, cu spațiu
 }
 return 0;
}

```

12. Scrieți un program **Python/C++** care citește din fișierul `jurnal.in` propoziții, scrise în fișier pe câte o linie. Programul scrie în fișierul `jurnal.out` doar propozițiile care conțin cel puțin o dată cuvântul „onoare” (indiferent de majuscule sau minuscule). Propozițiile sunt formate din cel mult 256 de caractere.

Exemplu:

| <u>jurnal.in</u>       | <u>jurnal.out</u>    |
|------------------------|----------------------|
| Am luptat cu onoare.   | Am luptat cu onoare. |
| Am câștigat o bătălie. | Onoarea este totul.  |
| Onoare si forta!       |                      |

### Limbajul Python

```
with open("jurnal.in", "r") as fin, open("jurnal.out", "w") as fout:
 for linie in fin:
 linie_strip = linie.strip()
 # căutăm cuvântul 'onoare' indiferent de majuscule
 if 'onoare' in linie_strip.lower().split():
 fout.write(linie)
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;

void toLower(char dest[], char sir[]) {
 int i = 0;
 strcpy(dest, sir);
 while (dest[i]!='\0') { dest[i] = tolower(dest[i]); i++; }
}

bool contine_onoare(char linie[]) {
 char linie_mica[300]; toLower(linie_mica, linie);
 return strstr(linie_mica, "onoare") != NULL;
}

int main() {
 ifstream fin("jurnal.in");
 ofstream fout("jurnal.out");
 char linie[300];
 while (fin.getline(linie, 300))
 if (contine_onoare(linie)) fout << linie << '\n';
 return 0;
}
```

13. Scrieți un program **Python/C++** care citește din fișierul `stele.in` un șir de `n` numere naturale ( $n < 1000$ ), fiecare reprezentând luminozitatea unei stele. Programul va scrie în `stele.out` toate valorile pare în ordine inversă.

Exemplu: `stele.in: 8 3 4 2 7 6; stele.out: 6 2 4 8`

### Limbajul Python

```
f = open("stele.in", "r")
linie = f.read()
f.close()
numere = linie.split()
i = len(numere) - 1
```

```

g = open("stele.out", "w")
while i >= 0:
 x = int(numere[i])
 if x % 2 == 0: g.write(str(x) + " ")
 i = i - 1
g.close()

```

### Limbajul C++

```

#include <iostream>
#include <fstream>
using namespace std;
int main() {
 ifstream fin("stele.in");
 ofstream fout("stele.out");
 int v[1000], n = 0, x;
 while (fin >> x)
 if (x % 2 == 0)
 v[n++] = x; // salvăm doar numerele pare
 for (int i = n - 1; i >= 0; i--) fout << v[i] << " ";
 return 0;
}

```

14. Scrieți un program Python/C++ care citește din fișierul `turn.in`,  $n$  numere. O cheie secretă este activată doar dacă cel mai mare număr prim din șir este egal cu suma tuturor celorlalte numere prime.

Programul va scrie în `turn.out` mesajul **CHEIE ACTIVATĂ** sau **EROARE**.

|         |                               |                        |
|---------|-------------------------------|------------------------|
| Exemplu | <code>turn.in:</code>         | <code>turn.out:</code> |
|         | 6                             | <b>CHEIE ACTIVATĂ</b>  |
|         | 3 5 7 2 17                    |                        |
|         | deoarece $17 = 3 + 5 + 7 + 2$ |                        |

### Limbajul Python

```

def este_prim(x):
 if x < 2: return False
 for d in range(2, int(x**0.5) + 1):
 if x % d == 0: return False
 return True
with open("turn.in", "r") as f:
 numere = list(map(int, f.read().split()))
maxim_prim = -1
suma = 0
for x in numere:
 if este_prim(x):
 if x > maxim_prim:
 maxim_prim = x
 suma += x
if maxim_prim != -1 and suma - maxim_prim == maxim_prim:
 mesaj = "CHEIE ACTIVATĂ"
else: mesaj = "EROARE"
with open("turn.out", "w") as g: g.write(mesaj)

```

## Limbajul C/C++

```
#include <fstream>
using namespace std;
bool estePrim(int x) {
 if (x < 2) return false;
 for (int d = 2; d * d <= x; ++d)
 if (x % d == 0) return false;
 return true;
}
int main() {
 ifstream fin("turn.in");
 ofstream fout("turn.out");
 int x, maxim = -1, suma = 0;
 while (fin >> x) {
 if (estePrim(x)) {
 if (x > maxim) maxim = x;
 suma += x;
 }
 if (maxim != -1 && suma - maxim == maxim) fout << "CHEIE ACTIVATĂ";
 else fout << "EROARE";
 }
 return 0;
}
```

15. Scrieți un program Python/C++ care citește din fișierul `carte.in` o matrice  $n \times m$  ( $n, m \leq 100$ ) cu temperaturi magice. Programul va scrie în `carte.out` temperatura maximă din fiecare coloană, fiecare pe câte o linie.

Exemplu `carte.in`:

```
3 3
1 5 2
3 8 6
4 4 7
```

`carte.out`:

```
4
8
7
```

## Limbajul Python

```
with open("carte.in", "r") as f:
 n, m = map(int, f.readline().split())
 matrice = [list(map(int, f.readline().split())) for _ in range(n)]
with open("carte.out", "w") as g:
 for col in range(m):
 maxim = matrice[0][col]
 for lin in range(1, n):
 if matrice[lin][col] > maxim:
 maxim = matrice[lin][col]
 g.write(str(maxim) + "\n")
```

## Limbajul C/C++

```
#include <fstream>
using namespace std;
int main() {
 ifstream fin("carte.in");
 ofstream fout("carte.out");
```

```

int n, m;
fin >> n >> m;
int a[101][101];
for (int i = 0; i < n; i++)
 for (int j = 0; j < m; j++) fin >> a[i][j];
for (int j = 0; j < m; j++) {
 int maxim = a[0][j];
 for (int i = 1; i < n; i++)
 if (a[i][j] > maxim) maxim = a[i][j];
 fout << maxim << '\n';
}
return 0;
}

```

16. Scrieți un program Python/C++ care citește din fișierul **simetrie.in** o matrice pătratică  $n \times n$ . Programul va scrie în **simetrie.out** mesajul SIMETRICA dacă matricea este simetrică față de diagonala principală, altfel va scrie NU ESTE SIMETRICA.

**Exemplu:**

|                     |                     |
|---------------------|---------------------|
| <b>simetrie.in:</b> | <b>simetrie.out</b> |
| 3                   | SIMETRICA           |
| 1 2 3               |                     |
| 2 4 5               |                     |
| 3 5 6               |                     |

### Limbajul Python

```

with open("simetrie.in", "r") as f:
 n = int(f.readline())
 mat = [list(map(int, f.readline().split())) for _ in range(n)]
 simetrica = True
 for i in range(n):
 for j in range(i):
 if mat[i][j] != mat[j][i]:
 simetrica = False
 break
 with open("simetrie.out", "w") as f:
 if simetrica: f.write("SIMETRICA")
 else: f.write("NU ESTE SIMETRICA")

```

### Limbajul C/C++

```

#include <fstream>
using namespace std;
int main() {
 ifstream fin("simetrie.in");
 ofstream fout("simetrie.out");
 int n;
 fin >> n;
 int a[100][100];
 for (int i = 0; i < n; i++)
 for (int j = 0; j < n; j++) fin >> a[i][j];

```

```

bool simetrica = true;
for (int i = 0; i < n && simetrica; i++) {
 for (int j = 0; j < i; j++)
 if (a[i][j] != a[j][i])
 { simetrica = false; break; }
}
if (simetrica)fout << "SIMETRICA";
else fout << "NU ESTE SIMETRICA";
return 0;
}

```

17. Scrieți un program Python/C++ care citește din fișierul **ecou.in** o matrice pătratică de dimensiune  $n \times n$ . Programul va scrie în fișierul **ecou.out** toate elementele de pe conturul matricei (prima și ultima linie, prima și ultima coloană), în ordinea parcurgerii în sensul acelor de ceasornic, pe o singură linie.

**Exemplu:**

| ecou.in     | ecou.out                     |
|-------------|------------------------------|
| 4           | 1 2 3 4 8 12 16 15 14 13 9 5 |
| 1 2 3 4     |                              |
| 5 6 7 8     |                              |
| 9 10 11 12  |                              |
| 13 14 15 16 |                              |

### Limbajul Python

```

with open("ecou.in") as f:
 n = int(f.readline())
 a = [list(map(int, f.readline().split())) for _ in range(n)]
rez = []
sus
for j in range(n):
 rez.append(a[0][j])
dreapta
for i in range(1, n - 1):
 rez.append(a[i][n - 1])
jos
for j in range(n - 1, -1, -1):
 rez.append(a[n - 1][j])
stanga
for i in range(n - 2, 0, -1):
 rez.append(a[i][0])
with open("ecou.out", "w") as f:
 f.write(" ".join(str(x) for x in rez))

```

### Limbajul C/C++

```

#include <fstream>
using namespace std;

int main() {
 ifstream fin("ecou.in");
 ofstream fout("ecou.out");

 int n;
 fin >> n;
 int a[100][100];

```

```

 for (int i = 0; i < n; i++) {
 for (int j = 0; j < n; j++) {
 fin >> a[i][j];
 }
 }
 // Prima linie
 for (int j = 0; j < n; j++) fout << a[0][j] << " ";
 // Ultima coloană (fără primul și ultimul element)
 for (int i = 1; i < n - 1; i++) fout << a[i][n - 1] << " ";
 // Ultima linie (de la dreapta la stânga)
 for (int j = n - 1; j >= 0; j--) fout << a[n - 1][j] << " ";
 // Prima coloană (de jos în sus, fără colțuri)
 for (int i = n - 2; i > 0; i--) fout << a[i][0] << " ";
 return 0;
}

```

18. Scrieți un program în **Python/C++** care citește din fișierul **cavaler.in** un număr natural **n**. Un cavaler poate trece poarta doar dacă numărul său are exact 3 divizori pozitivi (inclusiv 1 și el însuși). Programul va scrie în fișierul **cavaler.out** mesajul **POARTA SE DESCHIDE** dacă regula este respectată sau **RAMANE INCHISA** în caz contrar.

#### Limbajul Python

```

def este_prim(x):
 if x < 2:
 return False
 for d in range(2, int(x**0.5) + 1):
 if x % d == 0:
 return False
 return True

with open("cavaler.in", "r") as f:
 n = int(f.readline())

rad = int(n ** 0.5)
with open("cavaler.out", "w") as f:
 if rad * rad == n and este_prim(rad):
 f.write("POARTA SE DESCHIDE")
 else:
 f.write("RAMANE INCHISA")

```

#### Limbajul C/C++

```

#include <fstream>
#include <cmath>
using namespace std;

bool estePrim(int x) {
 if (x < 2) return false;
 for (int d = 2; d * d <= x; d++) {
 if (x % d == 0) return false;
 }
 return true;
}

```

```

int main() {
 ifstream fin("cavaler.in");
 ofstream fout("cavaler.out");

 int n;

 fin >> n;
 int rad = sqrt(n);
 if (rad * rad == n && estePrim(rad)) fout << "POARTA SE DESCHIDE";
 else fout << "RAMANE INCHISA";
 return 0;
}

```

19. Scrieți un program în **Python/C++** care citește din fișierul **stea.in** un număr natural **n** și șir de **n** numere naturale. Programul va determina și scrie în **stea.out** poziția elementului cu valoarea maximă. Dacă apar mai multe, se scrie doar prima.

**Exemplu:**

|                 |                 |
|-----------------|-----------------|
| <b>stea.in:</b> | <b>stea.out</b> |
| 7               | 2               |
| 3 9 2 9 1 5 7   |                 |

### Limbajul Python

```

with open("stea.in") as f:
 n = int(f.readline())
 nums = list(map(int, f.readline().split()))

max_val = nums[0]
max_pos = 1
for i in range(1, n):
 if nums[i] > max_val:
 max_val = nums[i]
 max_pos = i + 1 # poziție 1-based
with open("stea.out", "w") as f:
 f.write(str(max_pos))

```

### Limbajul C/C++

```

#include <fstream>
using namespace std;

int main() {
 ifstream fin("stea.in");
 ofstream fout("stea.out");
 int n, x, max_val = -1, max_pos = 0;
 fin >> n;
 for (int i = 1; i <= n; i++) {
 fin >> x;
 if (x > max_val) {
 max_val = x;
 max_pos = i; // salvăm prima poziție
 }
 }

 fout << max_pos;
 return 0;
}

```

20. Scrieți un program în **Python/C++** care citește din fișierul **regat.in** un număr natural **n** și un șir de **n** numere naturale. Programul va scrie în **regat.out** toate numerele care sunt egale cu suma cifrelor lor ridicate la puterea poziției lor în șir.

Exemplu: regat.in

3  
8 17 81

( $8^1 = 8$ ,  $8+1=9 \rightarrow 9^2=81$ )

regat.out:

8 81

### Limbajul Python

```
def suma_cifre(x):
 s = 0
 while x > 0:
 s += x % 10
 x //= 10
 return s
with open("regat.in") as f:
 n = int(f.readline())
 v = list(map(int, f.readline().split()))
rez = []
for i in range(n):
 suma = suma_cifre(v[i])
 if suma ** (i + 1) == v[i]: rez.append(str(v[i]))
with open("regat.out", "w") as f: f.write(" ".join(rez))
```

### Limbajul C/C++

```
#include <fstream>
#include <cmath>
using namespace std;
int suma_cifre(int x) {
 int s = 0;
 while (x)
 { s += x % 10; x /= 10; }
 return s;
}
int main() {
 ifstream fin("regat.in");
 ofstream fout("regat.out");
 int n, x;
 fin >> n;
 for (int i = 1; i <= n; i++) {
 fin >> x;
 int suma = suma_cifre(x);
 if (pow(suma, i) == x) fout << x << " ";
 }
 return 0;
}
```

21. Scrieți un program în Python/C++ care citește din fișierul **patrate.in** un număr natural  $n$  și un șir de  $n$  numere naturale ( $n \leq 100$ ). Programul va calcula suma pătratelor valorilor care sunt divizibile cu 3 și va scrie rezultatul în fișierul **patrate.out**.

Exemplu: patrate.in

5  
6 4 9 11 15

patrate.out

342

$$(6^2 + 9^2 + 15^2 = 36 + 81 + 225)$$

### Limbajul Python

```
Deschidem fișierul pentru citire
with open("pătrate.in", "r") as f: n_and_values = f.read().split()

n = int(n_and_values[0])
values = list(map(int, n_and_values[1:]))
suma = 0
for x in values:
 if x % 3 == 0: suma += x * x
Scriem rezultatul în fișierul de ieșire
with open("pătrate.out", "w") as f: f.write(str(suma))
```

### Limbajul C/C++

```
#include <fstream>
using namespace std;
int main() {
 ifstream fin("pătrate.in");
 ofstream fout("pătrate.out");
 int n, x; long long suma = 0;
 fin >> n;
 for (int i = 0; i < n; i++) {
 fin >> x;
 if (x % 3 == 0) suma += x * x;
 }
 fout << suma;
 return 0;
}
```

22. Fișierul **date.in** conține un șir de cel mult un milion de numere întregi din intervalul  $[-10^6, 10^6]$ , separate prin câte un spațiu. Scrieți un program Python/C++ care afișează pe ecran, în câte zerouri se termină produsul numerelor din fișier. Exemplu: dacă fișierul **date.in** conține numerele 20 25 32 7 6 125 35 36 se afișează pe ecran 7

### Limbajul Python

```
f=open('date.in','r')
cinci=0
doi=0
for line in f:
 for x_str in line.split():
 x=int(x_str)
 while x%5==0:
 cinci+=1
 x//= 5
 while x%2==0:
 doi+=1
 x//=2
if doi<cinci: print(doi)
else: print(cinci)
f.close()
```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("date.in");
int main()
{ int x,cinci=0,doi=0;
 while(f>>x)
 { while(x%5==0) { cinci++; x=x/5; }
 while(x%2==0) { doi++; x=x/2; } }
 if(doi<cinci) cout<<doi;
 else cout<<cinci;
 f.close();
 return 0;}
```

23. Fișierul `date.in` conține un șir de cel mult un milion de numere întregi din intervalul  $[-10^6, 10^6]$ , separate prin câte un spațiu. Scrieți un program Python/C++ care afișează pe ecran, mesajul "Da" dacă șirul din fișier este **zig-zag** sau mesajul "Nu" în caz contrar.

Un șir se numește zig-zag dacă oricare ar fi 3 elemente consecutive din șir nu sunt în ordine crescătoare sau descrescătoare.

Exemplu: dacă fișierul `date.in` conține numerele 20 25 32 7 6 125 35 36 se afișează pe ecran 7

## Limbajul Python

```
Python
f=open('date.in', 'r')
x=int(f.readline())
y=int(f.readline())

ok=0
nr=0
for line in f:
 for z_str in line.split():
 z=int(z_str)
 if (x<y and y>z) or (x>y and y<z):
 ok+=1
 x=y
 y=z
 nr+=1
if ok==nr: print("Da")
else: print("Nu")
f.close()
```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("date.in");
int main()
{ int x,y,z,ok=0,nr=0;
 f>>x;
```

```

f>>y;
while(f>>z)
{ if(x<y && y>z) ok++;
 if(x>y && y<z) ok++;
 nr++; x=y; y=z;
}
if(ok==nr) cout<<"Da";
else cout<<"Nu";
return 0;
}

```

24. Fișierul **date.in** conține un număr natural **n** urmat de **n** linii pe care se află perechi **(x,y)** de numere întregi din intervalul  $[-10^4, 10^4]$  ce reprezintă coordonatele unor puncte în planul XoY. Scrieți un program Python/C++ care afișează pe ecran pătratul razei celui mai mic cerc care conține toate punctele citite din fișier.

**Exemplu:** dacă fișierul **date.in** conține numerele

5

5 3

7 6

4 4

6 3

7 1

se afișează pe ecran 85

### Limbajul Python

```

with open('date.in', 'r') as f:
 n=int(f.readline())
 maxim=0
 for _ in range(n):
 line = f.readline()
 x_str, y_str = line.split()
 x=int(x_str)
 y=int(y_str)
 d=x*x +y*y
 if d>maxim:
 maxim=d
print(maxim)

```

### Limbajul C/C++

```

#include <iostream>
#include <fstream>
using namespace std;
ifstream f("date.in");
int main()
{ int n,x,y,i,d,maxim=0;
 f>>n;
 for(i=1;i<=n;i++)
 { f>>x>>y; d=x*x+y*y;
 if(maxim<d) maxim=d; }
 cout<<maxim;
 f.close(); return 0; }

```

25. Fișierul `date.in` conține un șir de litere mici din alfabetul englez, separate prin câte un spațiu. Scrieți un program Python/C++ care afișează pe ecran, care literă apare de cele mai multe ori. Dacă sunt mai multe litere care apar de număr maxim de ori se va afișa prima în ordine alfabetică.

Exemplu: dacă fișierul `date.in` conține lietererele `a e f s e f d a e f f r e t s e` afișează pe ecran `e`

#### Limbajul Python

```
with open("date.in", "r") as f:
 v=[0]*26
 maxim=0
 lit='a'
 nr=0
 for line in f:
 for x in line:
 nr+=1
 if nr%2==1: v[ord(x)-ord('a')]+= 1
 for i in range(26):
 if maxim<v[i]:
 maxim=v[i]
 lit=chr(i+ord('a'))
print(lit)
```

#### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("date.in");
int main()
{
 int v[26] = {0};
 int maxim = 0,i;
 char x,lit;
 while(f>>x) v[x-'a']++;
 for(i=0;i<26;i++)
 if(maxim<v[i]) {maxim=v[i]; lit=char(i+'a');}
 cout<<lit;
 f.close();
 return 0;
}
```

26. Fișierul `date.in` conține un număr de maxim 1.000.000 cifre. Scrieți un program Python/C++ care afișează pe ecran, mesajul `"Da"` numărul din fișier este divizibil cu 3 sau mesajul `"Nu"` în caz contrar.

**Exemplu:** dacă fișierul `date.in` conține numărul `324531231` se afișează pe ecran `Da`

#### Limbajul Python

```
with open("date.in", "r") as f:
 s=0
 while True:
 x=f.read(1)
 if not x: break
 if x.isdigit(): s+=int(x)
 if s%3==0: print("Da")
 else: print("Nu")
```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("date.in");
int main()
{
 char x;
 long long s=0;
 while(f.get(x))
 s=s+x-'0';
 if(s%3==0) cout<<"Da";
 else cout<<"Nu";
}
```

- 26) Fișierul `date.in` conține un număr de maxim 1000000 cifre. Scrieți un program Python/C++ care afișează pe ecran, mesajul "Nu" dacă cifrele numărului din fișier nu pot fi rearanjate astfel încât să formeze un număr palindrom sau dacă se pot aranja astfel încât să formeze un număr palindrom să afișeze cel mai mare număr care se poate forma.

**Exemplu:** dacă fișierul `date.in` conține numărul 2332113 se afișează pe ecran 3213123

## Limbajul Python

```
with open("date.in", "r") as f:
 v=[0]*10
 nr=0
 mij=-1
 data = f.read()
 for x in data:
 if x.isdigit():
 v[int(x)]+= 1
 for i in range(10):
 if v[i]%2==1:
 nr+=1
 mij=i
 v[i]-=1
 if nr>1:
 print("Nu")
 else:
 for i in range(9,-1,-1):
 for j in range(1,v[i]//2+1):
 print(i, end='')
 if mij!=-1:
 print(mij, end='')
 for i in range(10):
 for j in range(1,v[i]//2+1):
 print(i, end='')
 print()
```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("date.in");
int main()
{
 char x,nr=0;
 int v[10]={0},i,j,mij=-1;
 while(f.get(x)) v[x-'0']++;
 for(int i=0;i<=9;i++)
 if(v[i]%2==1) { nr++; mij=i; v[i]--; }
 if(nr>1) cout<<"Nu";
 else
 {
 for(i=9;i>=0;i--)
 for(j=1;j<=v[i]/2;j++) cout<<i;
 if(mij!=-1) cout<<mij;
 for(i=0;i<=9;i++) for(j=1;j<=v[i]/2;j++) cout<<i;
 }
}
```

28. Se definește șir **monoton-consecutiv** un șir de numere naturale nenule ordonate crescător, care au proprietatea că diferența, în valoare absolută, dintre oricare două numere situate pe poziții consecutive, este cel mult 1. Lungimea unui șir **monoton-consecutiv** este egală cu numărul de numere din șir. De exemplu, numerele 2, 2, 2, 3, 4, 4, 5 formează un șir **monoton-consecutiv**. Lungimea acestui șir **monoton-consecutiv** este 7. Un șir cu un singur element se consideră **monoton-consecutiv** de lungime 1.

Fișierul **date.in** conține un șir având cel mult  $10^6$  numere întregi din intervalul  $[-1000,1000]$ , separate prin câte un spațiu. Scrieți un program **Python/C++** care, printr-un algoritm eficient din punctul de vedere al timpului de executare, determină și afișează pe ecran numărul minim de șiruri **monoton-consecutive** care se pot forma, folosind toate numerele naturale nenule din fișier, după ordonarea crescătoare a acestora.

**Exemplu:** dacă fișierul conține numerele

3 7 3 2 -8 9 7 -7 6 1

atunci șirul se ordonează crescător -8 -7 1 2 3 3 6 7 7 9 și se pot forma, folosind toate numerele naturale nenule din fișier, minimum 3 șiruri **monoton-consecutive** 1 2 3 3, 6 7 7, 9, deci pe ecran se afișează valoarea 3.

## Limbajul Python

```
with open("date.in", "r") as fin:
 v=[0]*1001
 rez=0
 for val in fin.read().split():
 x=int(val)
 if x>0: v[x]=1
 for x in range(1,1001):
 if v[x]-v[x-1]>0: rez=rez+1
 print(rez)
```

## Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{ int v[1001]={0},x,rez=0;
 ifstream fin("date.in");
 while(fin>>x)
 if(x>0) // numar natural nenul
 v[x]=1;//se actualizeaza aparitia lui x in v
 for(x=1;x<=1000;x++)//se parcurge v
 if (v[x]-v[x-1]>0)//daca a aparut un nou sir
 rez=rez+1;
 cout<<rez;
 fin.close();
 return 0;
}
```

29. Fișierul `date.in` conține pe prima linie un număr natural nenul `n`, cu cel mult patru cifre, iar pe a doua linie un șir de cel puțin două și cel mult 100000 de numere naturale cu cel mult șase cifre. Numerele din șir sunt separate prin câte un spațiu.

Scrieți un program **Python/C++** care afișează, pe ecran, utilizând un algoritm eficient din punctul de vedere al timpului de executare, mesajul `da` dacă numerele din șir se pot rearanja astfel încât primele poziții să fie ocupate de primele `n` numere naturale, sau mesajul `nu`, în caz contrar

**Exemplu:** dacă fișierul `date.in` are conținutul

4

4 2 3 5 0 1 3

atunci pe ecran se afișează `da`

iar dacă fișierul `date.in` are conținutul

4

4 6 3 5 0 1 3

atunci pe ecran se afișează `nu`

## Limbajul Python

```
with open("date.in", "r") as fin:
 n=int(f.readline())
 v=[0]*10000
 ok=1
 for val in fin.read().split():
 x=int(val)
 if x<n: v[x]=1
 for x in range(n):
 if v[x]==0:
 ok=0
 if ok==1: print("da")
 else print("nu")
```

## Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
```

```

int main()
{ int v[10000]={0},n,x,ok=1;
 ifstream fin("date.in");
 fin>>n;
 while(fin>>x)
 if(x<n)
 v[x]=1;
 for(x=0;x<n;x++)
 if (v[x]==0)
 ok=0;
 if (ok==1)cout<<"da";
 else cout<<"nu";
 fin.close();
 return 0;
}

```

30. Fișierul **date.in** conține un șir având cel puțin 2 și cel mult 1000000 numere naturale din intervalul  $[0,10000]$ , separate prin câte un spațiu. Scrieți un program **Python/C++** care, printr-un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie, determină și afișează pe ecran mesajul **da** dacă șirul are proprietatea că între oricare două numere  $x$  și  $y$  din șir valoarea absolută (modulul) a diferenței lor este cel mult egală cu numărul de numere din șir, sau mesajul **nu**, în caz contrar.

**Exemplu:** dacă fișierul **date.in** conține numerele

4 4 3 3 4 3 2

atunci pe ecran se afișează mesajul **da**

iar dacă fișierul **date.in** conține numerele

4 6 8

atunci pe ecran se afișează mesajul **nu**

### Limbajul Python

```

with open("/date.in") as fin:
 maxi=-1
 mini=10001
 nr=0
 for val in f.read().split():
 x=int(val)

 nr=nr+1
 if x<mini: mini=x

 if x>maxi: maxi=x
 if maxi-mini<=nr: print("da")
 else: print("nu")

```

### Limbajul C/C++

```

#include<iostream>
#include<fstream>
using namespace std;
int main()
{ int x,mini=10001,maxi=-1,nr=0;
 ifstream fin("date.in");

```

```

while(fin>>x)
{
 nr++;
 if(x<mini) mini=x;
 if(x>maxi) maxi=x;}
if (maxi-mini<=nr) cout<<"da";
else cout<<"nu";
fin.close();
return 0;
}

```

31. Fișierul `date.in` conține un șir având cel puțin două și cel mult  $10^6$  numere întregi din intervalul  $[-10^6, 10^6]$ , separate prin câte un spațiu. Șirul conține cel puțin două numere naturale care sunt pătrate perfecte. Scrieți un program **Python/C++** care să afișeze, pe ecran, toți termenii șirului care sunt de tip pătrat perfect, în ordine descrescătoare, pe linii distincte. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul are conținutul

12 2 3 1 -1 2 5 16 36 1 5 0

se afișează pe ecran:

36  
16  
1  
1  
0

### Limbajul Python

```

with open("/date.in", "r") as fin:
 v=[0]*1001
 for val in fin.read().split():
 x=int(val)
 if x>=0:
 y=int(x**0.5)
 if y*y==x: v[y]=v[y]+1
 for y in range(1000,-1,-1):
 if v[y]>0:
 for x in range(0,v[y]): print(y*y)

```

### Limbajul C/C++

```

#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;
int main()
{
 ifstream fin("date.in");
 int x,y,v[1001]={0};
 while(fin>>x)
 if(x>=0)
 {y=int(sqrt(x));
 if(y*y==x) v[y]++; }
 for(y=1000;y>=0;y--)
 for(x=1;x<=v[y];x++) cout<<y*y<<endl;
 return 0;
}

```

32. Fișierul `date.in` conține minim două și maxim 1000000 de numere naturale separate prin câte un spațiu, formate din cel mult 6 cifre fiecare.

Scrieți un program **Python/C++** care citește numerele din fișier, apoi afișează, pe ecran, lungimea maximă a unei secvențe de numere naturale având exact două cifre, astfel încât una dintre cifrele numărului să fie triplul celeilalte, sau mesajul **nu exista**, dacă în fișier nu există niciun astfel de număr. Definim secvență de numere un șir de numere situate pe poziții alăturate în fișier, iar prin lungimea secvenței, numărul de numere din care este formată secvența. O secvență poate să aibă cel puțin lungimea 1 (atunci când are un singur element). Pentru rezolvarea cerinței, proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie.

**Exemplu:** dacă fișierul conține numerele

13 102 31 13 31 26 100 234 39 0

atunci pe ecran se afișează 4

iar dacă fișierul conține numerele

120 102 620 234 34 0

atunci pe ecran se afișează **nu exista**

### Limbajul Python

```
with open("/date.in", "r") as fin:
 lc=0
 lmax=0
 for val in fin.read().split():
 x=int(val)
 if x>=13 and x<=93:
 a=x//10
 b=x%10
 if a*3==b or b*3==a:
 lc=lc+1
 if lc>lmax: lmax=lc
 else: lc=0
 if lmax==0: print("nu exista")
 else: print(lmax)
```

### Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
 ifstream fin("date.in");
 int x,lc=0,lmax=0,a,b;
 while(fin>>x)
 if(x>=13 and x<=93)
 {
 a=x/10; b=x%10;
 if(a*3==b or b*3==a)lc++;
 if (lc>lmax)lmax=lc;
 }
 else lc=0;
 if(lmax!=0) cout<<lmax;
 else cout<<"nu exista";
 fin.close();
 return 0;
}
```

33. Fișierul `date.in` conține un șir având cel puțin două și cel mult un milion de numere întregi din intervalul  $[-10^6, 10^6]$ , separate prin câte un spațiu. Scrieți un program **Python/C++** care afișează, pe ecran, mesajul: **corect**, dacă în fișier sunt doar numere care au proprietatea că diferența dintre numărul de cifre a oricăror două numere situate pe poziții alăturate este  $-1$  sau  $1$ , respectiv mesajul **incorect**, în caz contrar. Programul utilizează un algoritm eficient din punctul de vedere al timpului de executare și al memoriei necesare. **Exemplu:** dacă fișierul `date.in` conține numerele `71 1 12 -1 41 927` se afișează pe ecran **corect**

iar dacă fișierul `date.in` conține numerele `11 1 12 -1 4` se afișează pe ecran mesajul **incorect** (deoarece numerele  $-1$  și  $4$  au același număr de cifre).

### Limbajul Python

```
with open("/date.in","r") as fin:
 nr=0
 ok=True
 for val in fin.readline().split():
 nr=nr+1
 if nr==1:
 a=val
 if a<0: a=-a
 ca=str(a)
 else:
 b=val
 if b<0: b=-b
 cb=str(b)
 if len(ca)-len(cb)!=1 and len(ca)-len(cb)!=-1: ok=False
 ca=cb
 if ok: print("corect")
 else: print("incorect")
```

### Limbajul C/C++

```
#include<iostream>
#include<fstream>
#include<cstdlib>
#include<cstring>
using namespace std;
int main()
{
 ifstream fin("date.in");
 int x,ca,cb,nr=0;
 char a[10],b[10];
 bool ok=true;
 while(fin>>x)
 {
 nr++;
 if(nr==1)
 {
 if(x<0) x=-x;
 itoa(x,a,10);
 ca=strlen(a);
 }
 else
 {
 if(x<0) x=-x;
 itoa(x,b,10);
```

```

 cb=strlen(b);
 if(ca-cb!=1 and ca-cb!=-1) ok=false;
 ca=cb;
 }
}
if(ok) cout<<"corect";
else cout<<"incorect";
fin.close();
return 0;
}

```

34. Fișierul `date.in` conține un șir având cel puțin două și cel mult un milion de numere naturale din intervalul  $[1, 10^6]$ . Numerele din șir sunt separate prin câte un spațiu. Scrieți un program **Python/C++** care citește numerele din fișier, apoi scrie în fișierul `date.out` cel mai mare număr impar care conține cifra 9 și cel mai mic număr par care conține cifra 0, pe linii distincte. Dacă nu există astfel de valori, se scrie în fișier mesajul `nu exista`. Pentru determinarea valorilor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**Exemplu:** dacă fișierul `date.in` conține numerele: 12 9 134 90 821 491 82 102

în fișierul `date.out` sunt scrise valorile:

```

491
90

```

Iar dacă fișierul `date.in` conține numerele: 12 9 134 901 821 491 82 12

în fișierul `date.out` este scris mesajul: `nu exista`

### Limbajul Python

```

fin=open("/date.in","r")
fout=open("/date.out","w")
maxi=0
minp=1000001
for val in fin.readline().split():
 a=int(val)
 ca=str(a)
 if a%2==0:
 if '0' in ca and a<minp: minp=a
 else:
 if '9' in ca and a>maxi: maxi=a
if maxi!=0 and minp!=1000001: fout.write(maxi,minp)
else: fout.write("nu exista")

```

### Limbajul C/C++

```

#include<fstream>
#include<cstdlib>
#include<cstring>
using namespace std;
int main()
{
 ifstream fin("date.in");
 ofstream fout("date.out");
 int x,maxi=0,minp=1000001;
 char s[10];

```

```

while(fin>>x)
{
 itoa(x,s,10);
 if(x%2==0) { if(strchr(s, '0') && x<minp) minp=x;}
 else
 { if(strchr(s, '9') && x>maxi) maxi=x;}
}
if(maxi!=0 && minp!=1000001) fout<<maxi<<endl<<minp;
else fout<<"nu exista";
fin.close(); fout.close();
return 0;
}

```

35. Fișierul **date.in** conține un șir având cel puțin două și cel mult un milion de numere naturale din intervalul  $[10^3, 10^8]$ . Numerele din șir sunt separate prin câte un spațiu. Scrieți un program **Python/C++** care citește numerele din fișier, apoi afișează, pe ecran, câte numere din fișier conțin în scrierea lor, respectând ordinea cifrelor, valoarea 2026. Dacă nu există astfel de valori, se scrie în fișier mesajul **nu exista**. Pentru determinarea valorilor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**Exemplu:**

dacă fișierul **date.in** conține numerele: 120263 2026 6206 20265 2000 32062  
 pe ecran se afișează valoarea: 3  
 Iar dacă fișierul **date.in** conține numerele: 1206 2602 6000 1202 8026  
 pe ecran se afișează mesajul: **nu exista**

### Limbajul Python

```

fin=open("/date.in","r")
cate=0
for val in fin.readline().split():
 s=str(val)
 if "2026" in s:
 cate=cate+1
if cate!=0:
 print(cate)
else:
 print("nu exista")

```

### Limbajul C/C++

```

#include<iostream>
#include<fstream>
#include<cstdlib>
#include<cstring>
using namespace std;
int main()
{
 ifstream fin("date.in");
 int x,cate=0;
 char s[10];
 while(fin>>x)
 {
 itoa(x,s,10);
 if(strstr(s, "2026")) cate++;
 }
 if(cate) cout<<cate;
 else cout<<"nu exista";
 fin.close();
 return 0;
}

```

36. Scrieți un program **Python/C++** care citește de la tastatură patru numere naturale  $n$ ,  $a$ ,  $b$ ,  $c$ , unde  $n$  este din intervalul  $[1, 1000]$ , iar  $a$ ,  $b$ ,  $c$  din intervalul  $[0, 9]$ . Programul scrie în fișierul `date.out` al  $n$ -lea termen al șirului definit astfel:

$f(1)=a, f(2)=b, f(3)=c,$

iar dacă  $n>3$  atunci  $f(n)=(f(n-1)+f(n-2)+f(n-3))\%10$

Pentru determinarea valorii cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**Exemplu:** dacă se citesc de la tastatură valorile: 12 1 3 2

în fișierul `date.out` este scrisă valoarea: 4

#### Limbajul Python

```
fout=open("/date.out","w")
n,a,b,c = map(int, input().split())
if n==1: fout.write(a)

if n==2: fout.write(b)

if n==3: fout.write(c)
if n>3:
 n=n-3
 while n>0:
 d=(a+b+c)%10
 a=b
 b=c
 c=d
 n=n-1
 fout.write(d)
fout.close()
```

#### Limbajul C/C++

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
 ofstream fout("date.out");
 int n,a,b,c,d;
 cin>>n>>a>>b>>c;
 if(n==1) fout<<a;
 if(n==2) fout<<b;
 if(n==3) fout<<c;
 if(n>3)
 {
 n=n-3;
 while(n)
 {
 d=(a+b+c)%10; a=b; b=c; c=d; n--;
 fout<<d;
 }
 fout.close();
 return 0;
 }
}
```

37. Scrieți un program **Python/C++** care citește de la tastatură trei numere naturale  $n, p, q$ , unde  $n$  este din intervalul  $[1, 30]$ , iar  $p, q$  din intervalul  $[1, 10]$ . Programul scrie în fișierul `date.out` numărul de cifre al celui de-al  $n$ -lea termen al șirului definit astfel:

$f(n) = p$ , dacă  $n=1$

$f(n) = q$ , dacă  $n=2$

iar dacă  $n > 2$  atunci  $f(n) = f(n-1) @ f(n-2)$ ,

unde  $@$  reprezintă operația de concatenare a lui  $f(n-1)$  cu  $f(n-2)$ .

Pentru determinarea valorii cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**De exemplu**, pentru  $n=6, p=3$  și  $q=5$ , primii 6 termeni ai șirului sunt 3, 5, 53, 535, 53553, 53553535. Deci în fișier se scrie valoarea 8 (deoarece 53553535 are 8 cifre).

| Limbajul Python                                                                                                                                                                                                                                                                                              | Limbajul C++                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> fout=open("/date.out","w") n,p,q = map(int, input().split()) if p&lt;10: a=1 else: a=2 if q&lt;10: b=1 else: b=2 if n==1:     fout.write(a) if n==2:     fout.write(b) if n&gt;2:     n=n-2     while n&gt;0:         c=a+b         a=b         b=c         n=n-1     fout.write(c) fout.close()</pre> | <pre> #include &lt;iostream&gt; #include &lt;fstream&gt; using namespace std; int main() {     ofstream fout("date.out");     int n,p,q,a,b,c;     cin&gt;&gt;n&gt;&gt;p&gt;&gt;q;     if(p&lt;10) a=1; else a=2;     if(q&lt;10) b=1; else b=2;     if(n==1) fout&lt;&lt;a;     if(n==2) fout&lt;&lt;b;     if(n&gt;2){ n=n-2;         while (n)         { c=a+b;             a=b;             b=c;             n--;}         fout&lt;&lt;c;}     fout.close();     return 0; }</pre> |

38. Scrieți un program **Python/C++** care citește din fișierul `stele.in` un șir de  $n$  numere naturale, fiecare reprezentând luminozitatea unei stele. Programul va scrie în `stele.out` **toate valorile impare în ordine crescătoare**.

**De exemplu**, pentru `stele.in` 8 3 4 2 7 6 , se va afișa în `stele.out` 3 7.

#### Limbajul Python

```

with open("stele.in", "r") as f:
 continut = f.read()
sir_numere = continut.split()
numere = [int(x) for x in sir_numere]
impare = []
```

```

for numar in numere:
 if numar % 2 == 1:
 impare.append(numar)
impare.sort()
with open("stele.out", "w") as f:
 for numar in impare:
 f.write(str(numar) + " ")

```

## Limbajul C/C++

```

#include <iostream>
#include <fstream>
#include <algorithm>
using namespace std;

int main() {
 ifstream fin("stele.in");
 ofstream fout("stele.out");
 int x, v[1000], k = 0;
 while (fin >> x) {
 if (x % 2 == 1) {
 v[k++] = x;
 }
 }
 sort(v, v + k);
 for (int i = 0; i < k; i++) {
 fout << v[i] << " ";
 }
 return 0;
}

```

39. Fișierul `date.in` conține un șir de cel puțin două și cel mult o mie de numere naturale din intervalul  $[0, 10^6]$ . Numerele sunt separate prin câte un spațiu.

Scrieți un program în **Python/C++** care citește șirul de numere din fișier și, utilizând un algoritm eficient din punctul de vedere al timpului de executare calculează produsul tuturor numerelor nenule din fișier și determină și afișează pe ecran câte dintre numere sunt egale cu zero. Dacă în fișier nu există niciun număr nenul, se va afișa mesajul **nu exista numere nenule**

**De exemplu**, pentru `date.in`: 0 3 -2 0 1 0 -1,  
 se va afișa:   **Produsul numerelor nenule: 6**  
                   **Numere egale cu zero: 3.**  
 pentru `date.in`: 0 0 0,  
 se va afișa :   **nu exista numere nenule**  
                   **Numere egale cu zero: 3.**

## Limbajul Python

```

with open("date.in", "r") as f:
 continut = f.read()
numere = list(map(int, continut.strip().split()))
produs = 1
are_nenule = False

```

```

numar_zero = 0
for x in numere:
 if x == 0: numar_zero += 1
 else:
 produs *= x
 are_nenule = True
if are_nenule: print("Produsul numerelor nenule:", produs)
else: print("nu exista numere nenule")
print("Numere egale cu zero:", numar_zero)

```

### Limbajul C/C++

```

#include <iostream>
#include <fstream>
using namespace std;
int main() {
 ifstream fin("date.in");
 long long x, produs = 1, numarZero = 0;
 bool areNenule = false;
 while (fin >> x) {
 if (x == 0) numarZero++;
 else { produs *= x; areNenule = true; }
 }
 if (areNenule) cout << "Produsul numerelor nenule: " << produs;
 else cout << "nu exista numere nenule";
 cout << "Numere egale cu zero: " << numarZero;
 return 0;
}

```

40. Scrieți un program **Python/C++** care citește din fișierul **date.in** două numere naturale **d1** și **d2** ( $0 \leq d1 \leq 9$ ,  $0 \leq d2 \leq 9$ ) și care afișează în fișier **date.out** al cincilea număr natural de **exact 4 cifre** pentru care diferența, în valoare absolută, dintre **prima și a doua cifră** este egală cu **d1** și diferența în valoare absolută dintre **a treia și a patra cifră** este egală cu **d2**. Dacă nu există un astfel de număr, se afișează: **imposibil**.

**De exemplu**, date.in 2 3 , se va afișa în date.out : 1529

### Limbajul Python

```

with open("date.in", "r") as f:
 d1, d2 = map(int, f.read().split())
gasite = 0
for numar in range(1000, 10000):
 mii = numar // 1000
 sute = (numar // 100) % 10
 zeci = (numar // 10) % 10
 unit = numar % 10
 if abs(mii - sute) == d1 and abs(zeci - unit) == d2:
 gasite += 1
 if gasite == 5:
 print(numar)
 break
if gasite < 5: print("imposibil")

```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;
int main() {
 ifstream fin("date.in");
 int d1, d2, gasite;
 fin >> d1 >> d2;
 gasite = 0;
 for (int numar = 1000; numar <= 9999; numar++) {
 int mii = numar / 1000;
 int sute = (numar / 100) % 10;
 int zeci = (numar / 10) % 10;
 int unit = numar % 10;
 if (abs(mii - sute) == d1 && abs(zeci - unit) == d2) {
 gasite++;
 if (gasite == 5) {
 cout << numar;
 return 0;
 }
 }
 }
 cout << "imposibil";
 return 0;
}
```

41. Scrieți un program **Python/C++** care citește din fișierul **date.in**, **n** numere naturale , printre care se consideră că există numerele impare în șir. Dacă **produsul celor mai mici două numere impare** este egal cu **diferența dintre cele mai mari două numere impare**, atunci se va scrie în fișierul de ieșire **date.out** mesajul „COD CONFIRMAT” sau „COD INVALID” dacă condiția nu este îndeplinită.

De exemplu, pentru **date.in** cu conținutul: 7

5 9 4 11 3 2 17

programul va scrie în **date.out**: COD CONFIRMAT

(Cele mai mici două: 3, 5 →  $3 \times 5 = 15$

Cele mai mari două: 17, 11 →  $17 - 11 = 6 \Rightarrow 15 \neq 6$ )

## Limbajul Python

```
with open("date.in", "r") as f:
 linii = list(map(int, f.read().split()))
 n = linii[0]
 numere = linii[1:]
 impare = [x for x in numere if x % 2 == 1]
 impare.sort()
 produs_minime = impare[0] * impare[1]
 diferenta_maxime = impare[-1] - impare[-2]
 if produs_minime == diferenta_maxime: mesaj = "COD CONFIRMAT"
 else: mesaj = "COD INVALID"
with open("date.out", "w") as f: f.write(mesaj)
```

```
#include <iostream>
#include <fstream>
#include <algorithm> // pentru sort
using namespace std;

int main() {
 ifstream fin("date.in");
 ofstream fout("date.out");
 int n, x, impare[1000], k = 0;
 fin >> n;
 for (int i = 0; i < n; i++) {
 fin >> x;
 if (x % 2 == 1) { impare[k] = x; k++; }
 }
 if (k < 4) {
 fout << "COD INVALID";
 return 0;
 }
 sort(impare, impare + k);
 int produs = impare[0] * impare[1];
 int diferenta = impare[k - 1] - impare[k - 2];
 if (produs == diferenta)
 fout << "COD CONFIRMAT";
 else
 fout << "COD INVALID";
 return 0;
}
```

42. Fișierul `date.in` conține un șir de cel puțin 10 și cel mult o mie de numere naturale, separate prin câte un spațiu. Fiecare număr are cel puțin **două** și cel mult **patru** cifre. Primul termen are numărul de ordine 1, al doilea are 2, și așa mai departe.

Scrieți un program în **Python/C++** care citește șirul de numere din fișier și, utilizând un algoritm eficient din punctul de vedere al timpului de executare, determină **numărul de ordine al aceluia termen care este precedat de cele mai multe valori având suma cifrelor divizibilă cu suma cifrelor sale**.

Dacă există mai mulți termeni cu această proprietate, se va afișa doar numărul de ordine al **primului** dintre ei.

**De exemplu**, dacă fișierul `date.in` conține numerele 123 456 789 234 321 111 222 333 147 258 369 741 852 963 159  
pe ecran se afișează 10

## Limbajul Python

```
def suma_cifrelor(n):
 suma = 0
 while n > 0:
 suma += n % 10
 n //= 10
 return suma

with open("date.in", "r") as f:
 continut = f.read()

numere_str = continut.strip().split()
n = len(numere_str)

numere = [int(numere_str[i]) for i in range(n)]
sume = [suma_cifrelor(numere[i]) for i in range(n)]

max_count = -1
poz_max = -1

for i in range(1, n):
 count = 0
 for j in range(i):
 if sume[j] % sume[i] == 0:
 count += 1
 if count > max_count:
 max_count = count
 poz_max = i + 1 # numerotare de la 1

print(poz_max)
```

## Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
const int MAX = 1000;

int sumaCifrelor(int n) {
 int suma = 0;
 while (n > 0) { suma += n % 10; n /= 10; }
 return suma;
}

int main() {
 ifstream fin("date.in");
 int numere[MAX], sume[MAX], n = 0;
 while (fin >> numere[n]) {
 sume[n] = sumaCifrelor(numere[n]); n++;
 }
 int maxCount = -1, pozMax = -1;
 for (int i = 1; i < n; i++) {
 int count = 0;
 for (int j = 0; j < i; j++)
 if (sume[j] % sume[i] == 0) count++;
 if (count > maxCount)
 { maxCount = count; pozMax = i + 1; }
 }
 cout << pozMax << endl;
 return 0;
}
```

43. Fișierul `date.in` conține un șir de cel puțin două și cel mult o mie de numere naturale din intervalul  $[0, 10^6]$ . Numerele sunt separate prin câte un spațiu, iar în șir există cel puțin doi termeni egali pe poziții consecutive.

Scrieți un program în **Python/C++** care citește șirul de numere din fișier și, utilizând un algoritm eficient din punctul de vedere al timpului de executare, determină poziția celui număr care este precedat de cele mai multe valori cu aceeași **paritate a sumei cifrelor** ca a sa (adică ambele sume fie pare, fie impare). Dacă există mai multe astfel de numere, se afișează **prima poziție**, iar dacă nu există niciunul cu astfel de valori înaintea lui, se va afișa 0. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei necesare și al timpului de executare.

**De exemplu**, dacă fișierul `date.in` conține numerele

12 45 78 23 66 89 11 14 22 30 27 33 42 57 60

se va afișa pe ecran : 14

deoarece pentru poziția 14 avem valoare 57=>sumă = 12 = par), sunt în față 66=>s=12, 11=>s=2, 22=>s=4, 33=>s=6, 42=>s=6) → 5 termeni cu sumă pară.

### Limbajul Python

```
def suma_cifrelor(n):
 suma = 0
 while n > 0:
 suma += n % 10
 n //= 10
 return suma

with open("date.in", "r") as f:
 sir = list(map(int, f.read().strip().split()))
n = len(sir)
paritati = [suma_cifrelor(x) % 2 for x in sir]
nr_pare = 0
nr_impere = 0
max_count = -1
poz_max = 0
for i in range(n):
 paritate = paritati[i]
 if paritate == 0: count = nr_pare
 else: count = nr_impere
 if count > max_count:
 max_count = count
 poz_max = i + 1
 if paritate == 0: nr_pare += 1
 else: nr_impere += 1
if max_count == 0: print(0)
else: print(poz_max)
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
```

```

int sumaCifrelor(int n) {
 int suma = 0;
 while (n > 0) {
 suma += n % 10;
 n /= 10;
 }
 return suma;
}

int main() {
 ifstream fin("date.in");
 const int MAX = 1000;
 int numere[MAX];
 int paritati[MAX]; // 0 pentru sumă pară, 1 pentru impară
 int n = 0;
 int x;
 while (fin >> x) {
 numere[n] = x;
 int suma = sumaCifrelor(x);
 paritati[n] = suma % 2; // 0 = par, 1 = impar
 n++;
 }

 int nr_pare = 0, nr_impere = 0;
 int max_count = -1;
 int poz_max = 0;
 for (int i = 0; i < n; i++) {
 int count;
 if (paritati[i] == 0) count = nr_pare;
 else count = nr_impere;
 if (count > max_count) {
 max_count = count;
 poz_max = i + 1; // numerotare de la 1
 }
 if (paritati[i] == 0) nr_pare++;
 else nr_impere++;
 }

 if (max_count == 0) cout << 0;
 else cout << poz_max;

 return 0;
}

```

44. Fișierul `date.in` conține pe prima linie un număr natural  $n$ , pe a doua linie un șir de  $n$  numere naturale **ordonate strict crescător**, numere naturale din intervalul  $[0, 10^6]$ , iar pe a treia linie un număr natural  $k$ .

Scrieți un program în **Python/C++** care citește datele de numere din fișier și, utilizând un algoritm eficient din punctul de vedere al timpului de executare, determină **și afișează poziția (indexul de la 1) primului număr mai mare decât  $k$  din șir**.

Dacă nu există un astfel de număr, se afișează mesajul `nu exista`

**De exemplu**, dacă fișierul `date.in` conține numerele:

```
7
3 8 10 15 21 30 42
20
```

se va afișa: 5(21 este primul număr > 20, iar poziția lui este 5)

Dacă fișierul `date.in` conține numerele:

```
5
10 20 30 40 50
60
```

se va afișa: `nu exista`

### Limbajul Python

```
with open("date.in", "r") as f:
 n = int(f.readline())
 linia_cu_numere = f.readline().split()
 k = int(f.readline())
 pozitie = 0
 gasit = False
 for i in range(n):
 val = int(linia_cu_numere[i])
 if val > k:
 pozitie = i + 1 # numerotare de la 1
 gasit = True
 break
 if gasit: print(pozitie)
 else: print("nu exista")
```

### Limbajul C/C++

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
 ifstream fin("date.in");
 int n, k, x, poz = 0;
 fin >> n;
 bool gasit = false;
 for (int i = 1; i <= n; ++i) {
 fin >> x;
 if (!gasit && x > k) { poz = i; gasit = true; break; }
 }
}
```

```

while (fin >> x) {
 k = x;
 break;
}
if (gasit) cout << poz;
 else cout << "nu exista";
return 0;
}

```

45. Fișierul `date.in` conține pe prima linie un număr natural  $n$ , pe a doua linie un șir de  $n$  numere naturale **ordonate strict crescător**, numere naturale din intervalul  $[0, 10^6]$ , iar pe a treia linie un număr natural  $k$ .

Scrieți un program în **Python/C++** care citește datele de numere din fișier și, utilizând un algoritm eficient din punctul de vedere al timpului de executare, determină **și afișează numărul de valori din șir care sunt multipli ai lui  $k$** . din șir. Dacă nu există un astfel de număr multiplu al lui  $k$  în șir, se afișează mesajul `nu exista`

**De exemplu**, dacă fișierul `date.in` conține numerele

```

7
100 90 80 70 60 50 40
20

```

se va afișa: 4

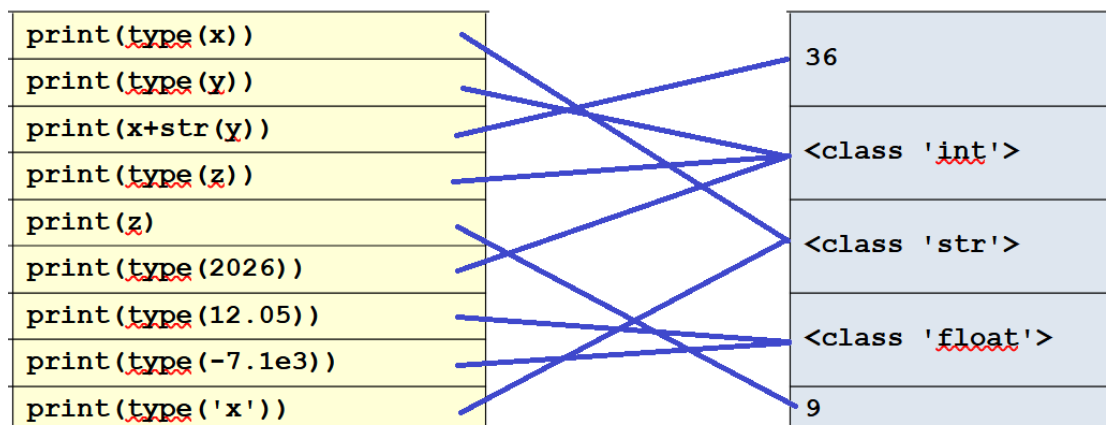
| Limbajul Python                                                                                                                                                                                                                                                                             | Limbajul C/C++                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> with open("date.in", "r") as f:     n = int(f.readline())     linie2 = f.readline().split()     k = int(f.readline()) multipli = 0 for s in linie2:     x = int(s)     if x % k == 0:         multipli += 1 if multipli == 0:     print("nu exista") else:     print(multipli) </pre> | <pre> #include &lt;iostream&gt; #include &lt;fstream&gt; using namespace std; int main() {     ifstream fin("date.in");     int n, x, k;     fin &gt;&gt; n;     int multipli = 0;     for (int i = 0; i &lt; n; ++i) {         fin &gt;&gt; x;         if (x % k == 0) multipli++;     }     fin &gt;&gt; k;     if (multipli == 0)         cout &lt;&lt; "nu exista";     else         cout &lt;&lt; multipli;     return 0; } </pre> |

## VIII. Răspunsuri

I.1.

- 1) c. 3) d. 5) c. 7) c. 9) d. 11) a. 13) c. 15) b.  
 2) a. 4) d. 6) b. 8) a. 10) a. 12) b. 14) a.

I.2.



I.3.

- a) 11 int      b) 32.5 float      c) True bool      d) 1 int      e) 16.0 float      f) 11 int

I.4.

| Constante numerice întregi | Constante numerice reale:     | Constante șir de caractere  |
|----------------------------|-------------------------------|-----------------------------|
| 2026, 0xB12, 0o34          | 142e-2, 723.45, 0.182, 2.25e1 | "xyz", '156', 'x', "523.26" |

I.5. a) -4); b) -5); c) -3); d) -1); e) -6); f) -2)

I.6. a) "abc12" str      b) 15 int      c) abcabcabc str      d) True bool

I.7.

1)a. 2)b. 3)d. 4)a. 5)c. 6)a. 7)a. 8)d. 9)b. 10)d. 11)d. 12)a.

I.8

1)a. 2)d. 3)d. 4)a. 5)a. 6)d. 7)b. 8)b. 9) d. 10)a.

V.1. m  
ame  
Exa  
amen  
Examen

V.2. s="CulegereInformatica"  
t="Culegere de Informatica"

V.3. Foarte\*Foarte\*BINE

V.4. c.

#### V.4.

|                                                                                                |  |                                       |
|------------------------------------------------------------------------------------------------|--|---------------------------------------|
| <pre>a = "Unu doi trei si patru" x = a.split() print(x)</pre>                                  |  | 102                                   |
| <pre>b = "Unu, doi, trei și patru" x = b.split(", ") print(x)</pre>                            |  | ['unu', 'doi', 'trei@patru']          |
| <pre>a="unu@doi@trei@patru" x = a.split("@") print(x)</pre>                                    |  | ['unu', 'doi', 'trei', 'patru']       |
| <pre>a="unu@doi@trei@patru" x = a.split("@",2) print(x)</pre>                                  |  | ['Unu', 'doi', 'trei și patru']       |
| <pre>s = "12 34 56" x = s.split() y = int(x[0]) z = int(x[1]) w = int(x[2]) print(y+z+w)</pre> |  | ['Unu', 'doi', 'trei', 'si', 'patru'] |

## IX. Bibliografie

- <https://www.python.org/>
- <https://www.pyml.ro/machine-learning-lectii-online-in-python.php>
- Vlad Tudor - Curs de programare in Python 3 - Fundamente pentru incepători, ebook, pdf, ISBN: 978-606-94898-1-9





**Editura L&S Soft | Infobits Academy**

**[www.infobits.ro](http://www.infobits.ro)**

**Cărți în format electronic cu specific informatic**

**<https://ebooks.infobits.ro/>**



***Curs online gratuit interactiv – limbajul Python 3***

**<https://www.pythonisti.ro/>**