

Algoritmi. Implementarea algoritmilor în C++.**Structura liniară. Aplicații.**

Structura liniară este formată doar din operații de **citire, scriere, atribuire**.

Aplicația 1. Gigel are **K** Gb liberi pe laptop și vrea să instaleze un joc care ocupă **P** Mb, știind că spațiul de memorie pe hard disc este suficient pentru asta. Câți Mb (Mega Bytes) îi rămân liberi, știind că 1 Gb = 1024 Mb?
Scrieți un algoritm care, pentru **K** și **P** date, calculează și afișează câți Mb rămân liberi după instalarea jocului.

| | |
|--|---|
| <pre>Date de intrare: K, P Date de iesire: Liber Date intermediare: z= cifra zecilor, u=cifra unitatilor Scrie "dati spatiul diponibil in Gb:" Citeste K Scrie "dati dimensiunea jocului i in Mb:" Citeste P K ← K * 1024 liber ← K - P scrie "spatiul liber ramas este " Scrie liber</pre> | <pre>#include <iostream> using namespace std; int main() { int K, P, liber; cout << "dati spatiul diponibil in Gb:"; cin >> K; cout << "dati dimensiunea joculu in Mb:"; cin >> P; K = K * 1024; liber = K - P; cout << "spatiul care va ramane liber = "; cout << liber << "\n"; return 0; }</pre> |
|--|---|

Împărțirea a două numere naturale **a** și **b** în C++ produce două valori: câtul și restul.

- a/b reprezintă câtul împărțirii întregi
- $a\%b$ reprezintă restul împărțirii întregi

Exemplu: dacă **a** = 14 și **b** = 5 atunci, **a/b** va fi **2** iar **a%b** va fi **4** pentru ca $13 = 5 * 2 + 4$)

Împărțirea unui număr natural **n** la 10 permite determinarea câtului și restului astfel:

- Restul împărțirii, **n%10** este cifra unităților (*Exemplu:* dacă **n** este 123 atunci, în urma evaluării expresiei **n%10**, se va obține valoarea **3**. Astfel, **n%10** reprezintă ultima cifră a numărului, cifra unităților.)
- Câtul împărțirii, **n/10** este numărul **n** din care lipsește cifra unităților (*Exemplu:* dacă **n** este 123 atunci, în urma evaluării expresiei **n/10**, se va obține valoarea 12)

Paritatea unui număr natural: un număr **n** este **par** dacă restul împărțirii la 2 este 0 (adică $n\%2$ este 0) și este **impar** dacă $n\%2 \neq 0$

✓ **Exerciții**

Ce rezultat se obține la evaluare următoarelor expresii?

- $125 \% 100 - 17 \% 2 + 20 / 4 = \dots$
- $n \% 10 + m \% 10 = \dots$, pentru $n = 3206$ și $m = 1000$
- $n \% 10 * 100 + n / 1000 = \dots$, pentru $n = 3206$
- $n / 100 \% 10 + n \% 10 * 100 = \dots$, pentru $n = 14598$

Aplicația 2. Se citește de la tastatură un număr **n**, de două cifre, diferite de 0. Să se construiască în **m** oglinditul acestuia apoi să se afișeze pe ecran.
Exemplu: dacă se citește pentru **n** valoarea **75**, se va afișa **m=57**.

Date de intrare: n
Date de iesire: ogl=oglinditul lui n
Date intermediare: z= cifra zecilor, u=cifra unitatilor

```

Scrie "dati n: "
Citeste n
u ← n%10
z ← n/10
oglindit ← u*10+z
scrie "oglinditul numarului n="
scrie oglindit

```

```

#include <iostream>
using namespace std;
int main()
{
    int n, z, u, oglindit;
    cout << "dati n: ";
    cin >> n;
    u = n % 10;
    z = n / 10;
    oglindit = u * 10 + z;
    cout << "oglinditul numarului n=";
    cout << oglindit << "\n";
    return 0;
}

```



Temă: Modificați algoritmul de mai sus pentru cazul în care n este format din trei cifre diferite de 0.

Exemplu: dacă se citește pentru **n** valoarea **125**, se va afișa **m=521**.

Aplicația 3. Se citește de la tastatură un număr **n**, de două cifre. Să se construiască în **rez** numărul format prin dublarea aparițiilor fiecărei cifre din n, apoi să se afișeze pe ecran. Exemplu: dacă se citește pentru **n** valoarea **75**, se va afișa **rez=7755**.

Date de intrare: n
Date de iesire: rez=noul numar obtinut
Date intermediare: z= cifra zecilor,
u=cifra unitatilor

```

Scrie "dati n: "
Citeste n
u ← n % 10
z ← n / 10
rez ← z * 1000 + z * 100 + u * 10 + u
Scrie "numarul obtinut prin dublarea
cifrelor="
scrie rez

```

```

#include <iostream>
using namespace std;
int main()
{
    int n, z, u, rez;
    cout << "dati n: ";
    cin >> n;
    u = n % 10;
    z = n / 10;
    rez = z * 1000 + z * 100 + u * 10 + u;
    cout << "numarul obtinut prin dublarea cifrelor=";
    cout << rez << "\n";
    return 0;
}

```



Temă: Modificați algoritmul de mai sus pentru a adăuga câte o cifră **0** după fiecare cifră a lui n. Exemplu: dacă se citește pentru **n** valoarea **12**, se va afișa **m=1020**.