

C++ Lecția 4

Structurile repetitive. Introducere.

1. Restul împărțirii la 10

- împărțind un număr întreg la 10 obținem un **cât** și un **rest**
- **restul** împărțirii la 10 indică **întotdeauna ultima cifră** a numărului

Exemplu:

Operație	Rezultat
1414 1 % 10	1
33 2 % 10	2
100 0 % 10	0
1990 9 % 10	9
$\overline{\sigma_0 \sigma_1 \sigma_2 \sigma_3 \dots \sigma_{k-1} \sigma_k}$ reprezintă un număr de $(k + 1)$ cifre, unde σ_0 este prima cifră a numărului și σ_k este ultima {0 <= k}	σ_k

2. Câtul împărțirii la 10

- **câtul** împărțirii la 10 indică **întotdeauna numărul FĂRĂ ultima cifră**

Operație	Rezultat
1414 1 / 10	1414
33 2 / 10	33
100 0 / 10	100
(0)9 / 10	0
$\overline{\sigma_0 \sigma_1 \sigma_2 \sigma_3 \dots \sigma_{k-1} \sigma_k}$ reprezintă un număr de $(k + 1)$ cifre, unde σ_0 este prima cifră a numărului și σ_k este ultima {0 <= k}	$\overline{\sigma_0 \sigma_1 \sigma_2 \sigma_3 \dots \sigma_{k-1}}$

C++ Lecția 4

- Dacă numărul conține o singură cifră, atunci câtul împărțirii la 10 va fi 0 (numărul rămâne fără nicio cifră)
- $0 / 10$ va fi tot 0

Concluzie:

Operație	Rezultat
$c = n \% 10$	variabila c conține ultima cifră a lui n
$n = n / 10$	se elimină ultima cifră a lui n

3. Suma cifrelor unui număr

- a. Dacă numărul are o singură cifră, suma acestor "cifre" este trivială

Exemplu:

3 -> 3

4 -> 4

n -> n

- b. Dacă numărul are **două** cifre, suma acestor cifre va fi:

ultima_cifra + prima_cifra

Exemplu:

33 -> $3 + 3 = 6$

45 -> $5 + 4 = 9$

95 -> $5 + 9 = 14$

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

int main() {
    int n, sum = 0;
    cout << "Introduceti un numar format din doua cifre: ";
    cin >> n;

    sum = n % 10 + n / 10;

    cout << "Suma cifrelor numarului este: " << sum << endl;
    return 0;
}
```

C++ Lecția 4

c. Dacă numărul are 3 cifre, pe același raționament, suma acestora va fi:

$$\text{ultima_cifra} + \text{a_doua_cifra} + \text{prima_cifra}$$

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

int main() {
    int n, sum;
    cout << "Introduceți un număr format din trei cifre: ";
    cin >> n;
    sum = n % 10 + (n / 10) % 10 + (n / 100);
    cout << "Suma cifrelor numărului este: " << sum << endl;
    return 0;
}
```

d. Cu cât crește numărul de cifre, devine tot mai greu să accesăm cifrele.

- Trebuie să găsim o cale pentru a accesa cifrele **indiferent de numărul acestora**
- Soluția este să repetăm grupul de instrucțiuni care:
 - extrage ultima cifră
 - elimină ultima cifră

```
Citire n
s <- 0
Repetă
    s <- s + (ultima cifră a lui n)
    n <- (n fără ultima cifră)
PânăCând n = 0
Afișare s
```

Astfel, parcurgem toate cifrele numărului de la stânga la dreapta, până când acesta rămâne fără cifre (n = 0)

C++ Lecția 4

- Exemplu: $n = 32310$

Instrucțiune	n	s
Citire n	32310	-
s <- 0	32310	0
@Repetă	32310	0
s <- s + (ultima cifră a lui n)	3231 0	0 + 0 = 0 adăugăm cifra la sumă
n <- (n fără ultima cifră)	3231 eliminăm ultima cifră	0
ContinuăRepetă? DA (n nu este = cu 0)	3231	0

@

s <- s + (ultima cifră a lui n)	323 1	0 + 1 = 1 adăugăm cifra la sumă
n <- (n fără ultima cifră)	323 eliminăm ultima cifră	1
ContinuăRepetă? DA (n nu este = cu 0)	323	1

@

s <- s + (ultima cifră a lui n)	32 3	1 + 3 = 4 adăugăm cifra la sumă
n <- (n fără ultima cifră)	32 eliminăm ultima cifră	4
ContinuăRepetă? DA (n nu este = cu 0)	32	4

@

s <- s + (ultima cifră a lui n)	32	4 + 2 = 6 adăugăm cifra la sumă
n <- (n fără ultima cifră)	32 -> 3 eliminăm ultima cifră	6
ContinuăRepetă? DA (n nu este = cu 0)	3	6

@

s <- s + (ultima cifră a lui n)	3	6 + 3 = 9 adăugăm cifra la sumă
n <- (n fără ultima cifră)	0 eliminăm ultima cifră	9
ContinuăRepetă? NU (n = 0)	0	9
FINAL	0	9

C++ Lecția 4

Soluție în c++

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

int main() {
    int n, sum = 0;
    cout << "Introduceți un număr: ";
    cin >> n;

    while (n != 0) {
        sum = sum + (n % 10);
        n = n / 10;
    }

    cout << "Suma cifrelor numărului este: " << sum << endl;
    return 0;
}
```

- Tot ce se află între acoladele lui **while** se va repeta **cât timp** condiția din paranteze ($n \neq 0$) se îndeplinește
- La fiecare *iterație* se verifică acea condiție ($n \neq 0$)
 - dacă ea se îndeplinește, **se continuă** repetiția
 - dacă nu se îndeplinește, **se oprește** repetiția și se trece mai departe
- Cu alte cuvinte, instrucțiunile din while se repetă:
 - **până când $n = 0$**
(sau)
 - **cât timp $n \neq 0$**
- **Cele două exprimări sunt echivalente, diferă condiția (care este negată)**
- În traducere liberă, se actualizează suma și se elimină ultima cifră a lui n cât timp n nu este 0

4. Aplicații

1. Afișați toate numerele de la 0 la n, unde n este un număr întreg citit de la tastatură

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

int main() {
    int n;
    cout << "Introdu un număr: "; // cere utilizatorului să introducă un număr
    cin >> n;

    int i = 0; // inițializează variabila i cu 0
    while (i <= n) {
        cout << i << endl; // afișează valoarea curentă a lui i
        i++; // adaugă 1 la i
    }
    return 0;
}
```

2. Afișați toate numerele pare de la 0 la n, unde n este un număr întreg citit de la tastatură.

- **Varianta 1** (cu **if**)

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

int main() {
    int n;
    cout << "Introdu un număr: "; // cere utilizatorului să introducă un număr
    cin >> n;

    int i = 0; // inițializează variabila i cu 0
    while (i <= n) {
        if (i % 2 == 0) { // verifică dacă i este par
            cout << i << endl; // afișează valoarea curentă a lui i
        }
        i++; // adaugă 1 la i
    }
    return 0;
}
```

- **Varianta 2 (fără if)**
- deoarece numerele pare merg din 2 în 2, este suficient să creștem valoarea variabilei **i** cu 2 la fiecare pas
- i va fi mereu de forma:
 $0 \dots 2 \dots 4 \dots 6 \dots \dots 2k, k \in N \text{ (nr natural)}$

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

int main() {
    int n;
    cout << "Introdu un număr: "; // cere utilizatorului să introducă un număr
    cin >> n;

    int i = 0; // inițializează variabila i cu 0
    while (i <= n) {
        cout << i << endl; // afișează valoarea curentă a lui i
        i = i + 2; // adaugă 2 la i
    }
    return 0;
}
```

3. Calculează suma cifrelor pare ale unui număr n citit de la tastatură

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

int main() {
    int n; // numărul introdus de la tastatură
    int suma = 0; // variabila pentru a stoca suma cifrelor pare
    cout << "Introdu un număr: ";
    cin >> n;

    while (n > 0) { // cât timp numărul mai are cifre
        int cifra = n % 10; // extrage ultima cifră din număr
        if (cifra % 2 == 0) { // verifică dacă cifra este pară
            suma = suma + cifra; // adaugă cifra la suma
        }
        n = n / 10; // elimină ultima cifră din număr
    }
    cout << "Suma cifrelor pare este: " << suma << endl;
    return 0;
}
```

C++ Lecția 4

4. Calculează suma cifrelor impare ale unui număr **n** citit de la tastatură
5. Determină numărul de cifre ale lui **n** egale cu o cifră **c** dată.

Exemplu:

```
n = 124474, c = 4 -> 3
n = 12444, c = 2 -> 1
n = 12, c = 3 -> 0 (cifra 3 nu apare niciunde
                        în număr)
```