

Fundamentele Programarii – Laborator 2

Reprezentarea numerelor in calculator

Obiective

La finalul acestui laborator, studentul va fi capabil sa:

- intelega modul de reprezentare a numerelor intregi in calculator;
- faca diferenta intre tipurile **signed** si **unsigned**;
- explice notiunea de overflow;
- utilizeze corect tipurile de date intregi in C++.

Notiuni teoretice

Reprezentarea numerelor

Calculatorul memoreaza datele sub forma de **biti**. Un bit poate avea una din valorile:

$$0 \text{ sau } 1$$

Un grup de 8 biti formeaza un **byte**.

Numere intregi fara semn (**unsigned**)

Un numar intreg fara semn pe n biti poate reprezenta valori din intervalul:

$$[0, 2^n - 1]$$

Exemplu: pe 8 biti:

$$[0, 255]$$

Numere intregi cu semn (**signed**)

Pentru numerele cu semn se foloseste reprezentarea in **complement fata de 2**.

Pentru n biti, intervalul este:

$$[-2^{n-1}, 2^{n-1} - 1]$$

Exemplu: pe 8 biti:

$$[-128, 127]$$

Complement fata de 2

Pentru a obtine reprezentarea unui numar negativ:

1. Se scrie valoarea absoluta in binar;
2. Se inverseaza bitii (complement fata de 1);
3. Se aduna 1.

Exemplu: Reprezentarea lui -5 pe 8 biti:

$$5 = 00000101$$

$$\text{complement 1} = 11111010$$

$$\text{complement 2} = 11111011$$

Overflow

Definitie

Overflow-ul apare atunci cand rezultatul unei operatii depaseste intervalul de valori ce poate fi reprezentat de un tip de date.

Exemplu:

$$127 + 1 = -128 \quad (\text{pentru signed char})$$

Tipuri de date intregi in C++

Tip	Dimensiune tipica
char	1 byte
short	2 bytes
int	4 bytes
long	4 sau 8 bytes
long long	8 bytes

Exemple C++

```
1      #include <iostream>
2      using namespace std;
3
4      int main() {
5          unsigned char a = 255;
6          a = a + 1;
7          cout << (int)a << endl;
8
9          signed char b = 127;
10         b = b + 1;
11         cout << (int)b << endl;
12     }
```

```
13         return 0;
14     }
```

Exercitii propuse

Exercitii de baza

1. Care este intervalul de valori pentru:
 - `unsigned char`
 - `signed char`
2. Scrieti reprezentarea binara pe 8 biti pentru:

12, -12, 64, -64

Exercitii aplicative

1. Scrieti un program care evidentiaza overflow-ul pentru tipul `signed char`.
2. Ce se intampla daca adunam 1 la valoarea maxima a unui `unsigned char`?

Exercitii bonus

1. Determinati valoarea zecimala a reprezentarii binare:

11110110

interpretata ca `signed char`.

2. Explicati de ce overflow-ul este periculos in programare.

Observatii

Alegerea corecta a tipului de date este esentiala pentru evitarea erorilor logice si a comportamentelor neasteptate.

Laboratorul urmator: Operatii de intrare/iesire. Fluxuri