

Programarea calculatoarelor

Instrucțiuni în limbajul C

Profesor: Maria Guțu

Instrucțiuni decizionale în C

- **Instrucțiunea IF;**
- **Instrucțiunea IF-ELSE;**
- **Instrucțiunea IF imbricată;**
- **Instrucțiunea Switch;**
- **Operatorul condiționat.**

Instrucțiunea decizională IF

Sintaxa:

if(expression)

program statement

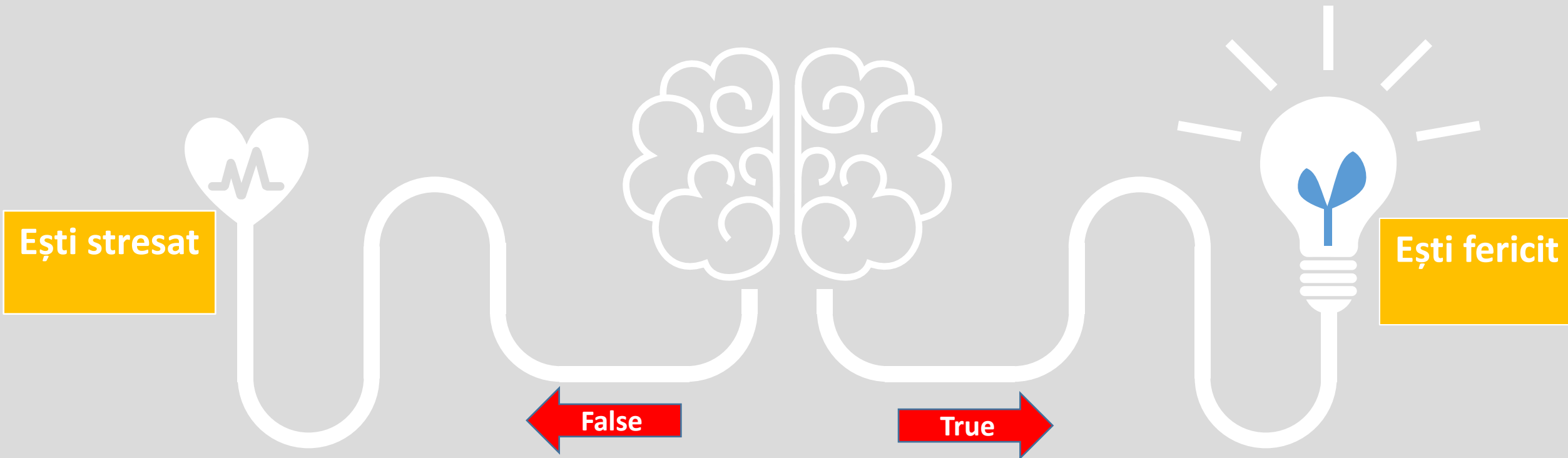
Semantica:

If (plouă)

iau umbrela

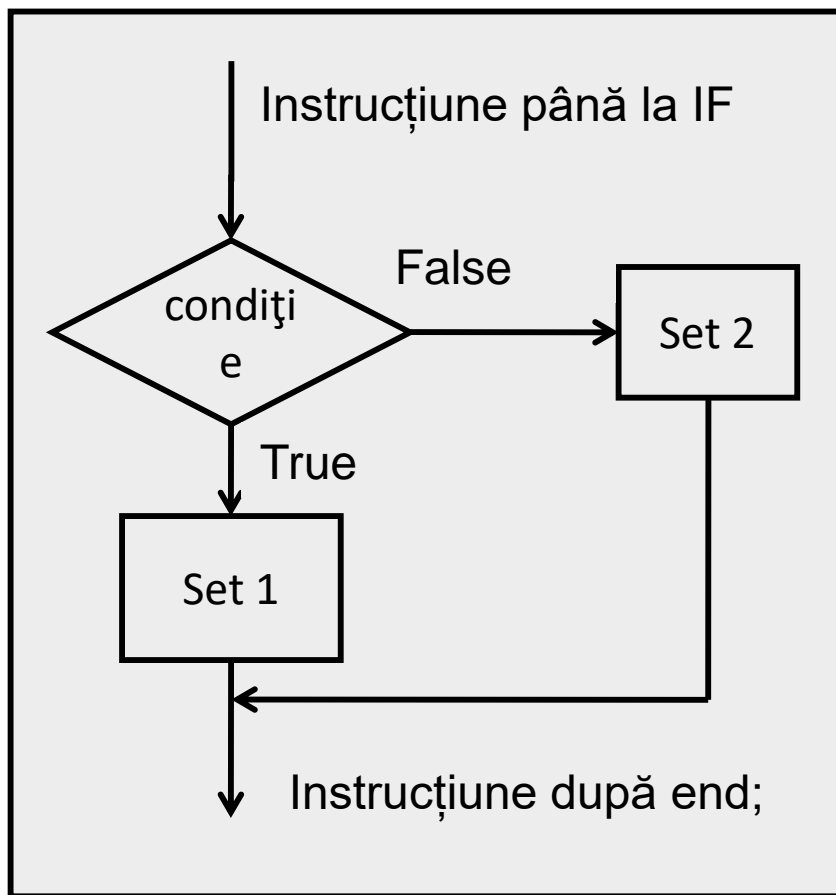
Instrucțiunea IF

Dacă ai învățat și ești pregătit de lecție

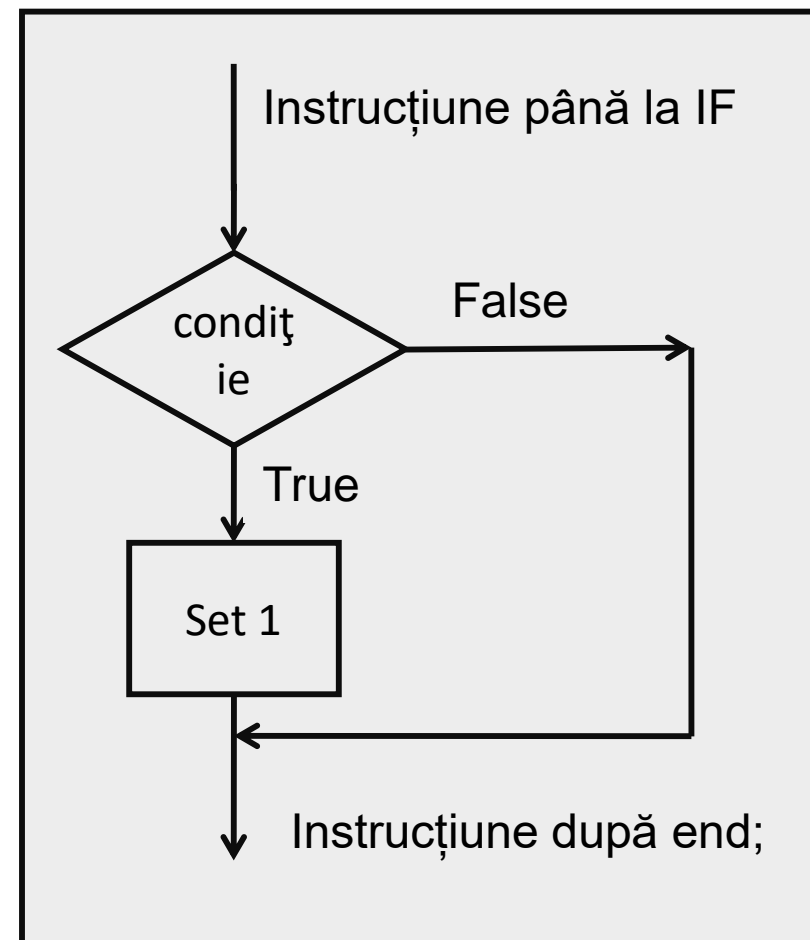


Condiții

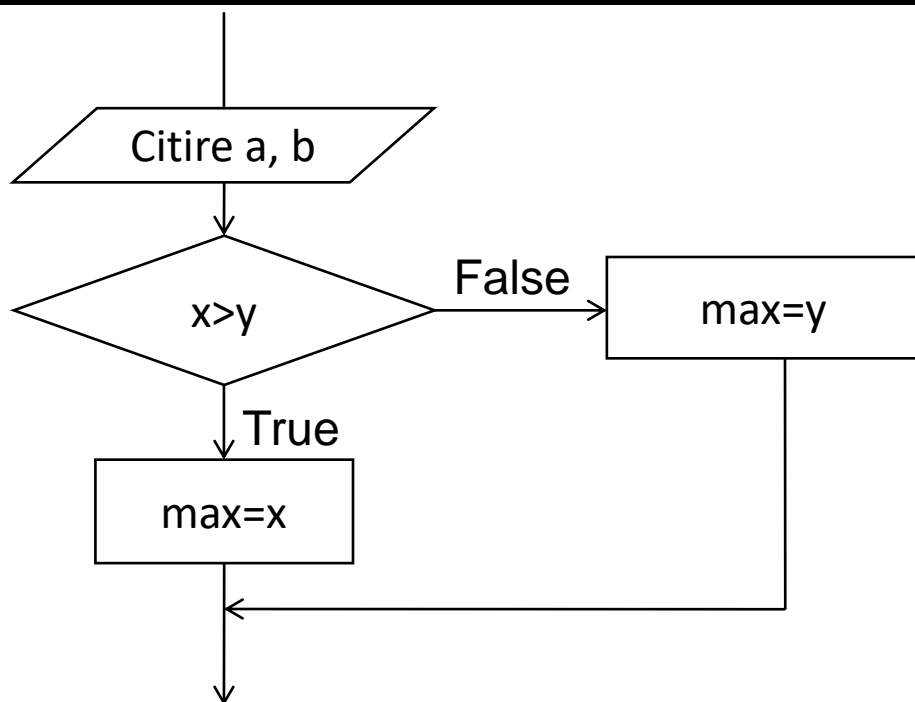
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`
- Equal to: `a == b`
- Not Equal to: `a != b`



Dacă rezultatul evaluării *Condiției* este *true*, atunci se execută *Set1* iar, dacă *Condiția* are valoare *false*, atunci se execută *Set2*, dacă există, în caz contrar se trece la următoarea secvență de cod din program.



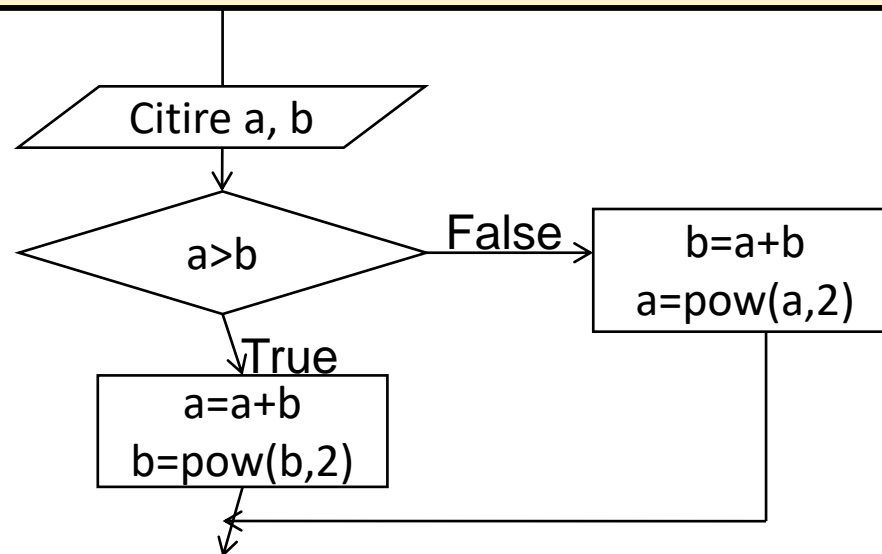
Se consideră două numere întregi distincte. Să se scrie un program care determină numărul mai mare.



```
#include <stdio.h>
int main()
{
    int x, y, max;
    scanf("%d%d",&x,&y);
    if (x>y) max = x;
        else max = y;
    printf("El.max: %d",max);
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>
int main()
{
    int a, b;
    scanf("%d%d",&a,&b);
    if (a>b) { a = a+b;
              b = pow(b,2);
            } else { b = a+b;
                   a = pow(a,2);
            }
    printf("a=%d; b=%d",a, b);
    return 0;
}
```

Se consideră două numere întregi distincte. Să se scrie un program care înlocuiește numărul mai mare cu suma numerelor date, iar numărul mai mic – cu pătratul său.




```
#include <stdio.h>
int main()
{
    int time = 22;
    if (time < 10) {
        printf("Good morning.");
    } else if (time < 20) {
        printf("Good day.");
    } else {
        printf("Good evening.");
    }

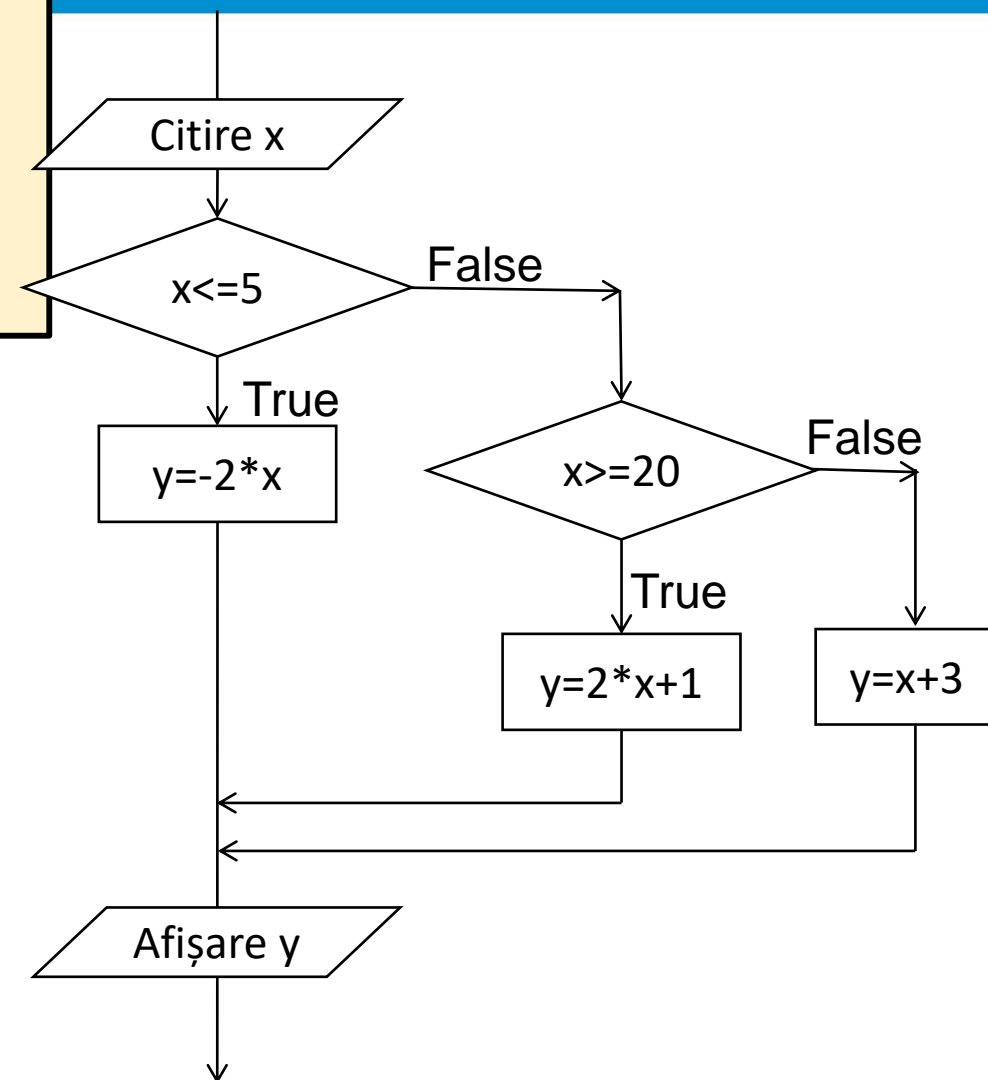
    return 0;
}
```

Exemplu 3

Să se calculeze valoarea funcției y definită pe mulțimea numerelor reale, pentru un număr x citit de la tastieră.

$$y = \begin{cases} -2x, & x \leq 5 \\ x + 3, & 5 < x < 20 \\ 2x + 1, & x \geq 20 \end{cases}$$

```
#include <stdio.h>
int main(){
    int x, y;
    scanf("%d",&x);
    if (x<=5) y = -2*x;
        else if (x>=20) y = 2*x+1;
            else y = x+3;
    printf("y=%d",y);
    return 0;}
```



Determinarea dacă un an este bisect sau nu

```
#include <stdio.h>
int main()
{
    int an;
    scanf("%d", &an);
    if(an % 400 == 0 || (an % 4 == 0 && an % 100 != 0))
        printf(" este un an bisect\n");
    else
        printf(" nu este un an bisect\n");
    return 0;
}
```



Instrucțiunea decizională IF

Instrucțiunea **IF** este utilizată pentru a stabili execuția unei instrucțiuni de program (sau declarații dacă sunt incluse în paranteze) pe baza unor condiții specificate.

Instrucțiunea **if / else if /else** se folosește pentru a implementa structuri decizionale cu mai mult de două alternative și se numesc **instrucțiuni imbricate**.

Instrucțiunea decizională Switch

Forma generală:

```
switch (opțiune) {  
    case c1: instructiuni 1; break;  
    case c2: instructiuni 2; break;  
    .....  
    case cn: instructiuni n; break;  
    default: instructiuni; break;  
}
```

Expresia **opțiune** aflată între paranteze este comparată succesiv cu valorile **c1**, **c2...**, **cn**, care *trebuie să fie ori constante simple ori expresii constante*.

Dacă se găsește un caz a cărui valoare este egală cu valoarea din **opțiune**, urmează instrucțiunile de program care execută cazul.

Nu uitați să includeți declarația de pauză break la sfârșitul fiecărui caz. Dacă uitați să faceți acest lucru pentru un caz particular execuția programului va continua cu următorul caz, ori de câte ori acest caz va fi executat.

Instrucțiunea decizională Switch

Forma generală:

```
switch (opțiune) {  
    case c1: instructiuni 1; break;  
    case c2: instructiuni 2; break;  
    .....  
    case cn: instructiuni n; break;  
    default: instructiuni; break;  
}
```

Cazul opțional special numit **default** este executat *dacă valoarea expresiei nu se potrivește cu niciuna din valorile cazului.*

De îndată ce este întâlnită instrucțiunea **break**; într-un ciclu, iterațiile ciclului se opresc în punctul unde se întâlnește **break**; și controlul revine din ciclu imediat la prima declarație de după ciclu.



Exemplu 1

```
// Program to create a simple calculator  
#include <stdio.h>
```

```
int main() {  
    char operation;  
    double n1, n2;
```

```
    printf("Enter an operator (+, -, *, /): ");  
    scanf("%c", &operation);  
    printf("Enter two operands: ");  
    scanf("%lf %lf", &n1, &n2);
```

```
    switch(operation) {  
        case '+':  
            printf("%.1lf + %.1lf = %.1lf", n1, n2, n1+n2); break;  
        case '-':  
            printf("%.1lf - %.1lf = %.1lf", n1, n2, n1-n2); break;  
        case '*':  
            printf("%.1lf * %.1lf = %.1lf", n1, n2, n1*n2); break;  
        case '/':  
            printf("%.1lf / %.1lf = %.1lf", n1, n2, n1/n2); break;  
        // operator doesn't match any case constant +, -, *, /  
        default:  
            printf("Error! operator is not correct");  
    }  
    return 0;  
}
```



```
#include <stdio.h>
int main() {
    int num;
    printf("Input a one-digit number\n");    scanf("%d",&num);
        switch (num) {
            case 0: case 2: case 4: case 6: case 8:
                printf("%d - an even number\n", num); break;
            case 1: case 3: case 5: case 7: case 9:
                printf("%d - an odd number\n", num); break;
            default:
                printf("%d not a one-digit number \n", num); break;
        }
    return 0;
}
```




Exemplu 3

```
#include <stdio.h>
int main(){
    char ch;
    printf("Input a character\n");
    scanf("%c",&ch);
```

```
switch (ch) {
    case 'A' ... 'Z':
        printf("%c in range A to Z\n", ch);
        printf("%d in range %d to %d\n", ch, 'A', 'Z');
        break;
    case 'a' ... 'z':
        printf("%c in range 'a' to 'z'\n", ch);
        printf("%d in range %d to %d\n", ch, 'a', 'z');
        break;
    default:
        printf("%d not in range\n", ch);
        break;
}
return 0; }
```



Un exemplu de utilizare a instrucțiunii **SWITCH** atunci când avem un interval prestabilit de valori.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(0));
    int num = rand() % 100 + 1;
```

Reține:

Corect - **case 1 ... 5:**

// cele trei puncte sunt precedate și
succedate de câte un spațiu

Greșit - **case 1...5:**

```
switch (num) {
    case 1 ... 10:
        printf("%d in range 1 to 10\n", num);
        break;
    case 19 ... 60:
        printf("%d in range 19 to 60\n", num);
        break;
    default:
        printf("%d not in range\n", num);
        break;
}
return 0;
}
```



Instrucțiuni Repetitive

Instrucțiunea FOR

Instrucțiunea While

Instrucțiunea Do...While



Aplicații Practice!!!

(maria.gutu@iis.utm.md)