

## Laborator 9

### Funcții

#### Exemplu 1:

```
#include <stdio.h>

/* definitia functiei */
int suma (int x, int y)
{
    return x+y;
}

int main()
{
    int a,b;
    printf("Dati doua numere:");
    scanf("%d%d",&a,&b);
    /* apelul functiei */
    printf("%d + %d = %d", a, b, suma(a,b));
    return 0;
}
```

#### Exemplu 2: citirea și afișarea unui vector de întregi

Fie un vector cu maxim 50 de elemente întregi. Să se scrie un program care afișează și prelucrează meniul:

C.Citire  
A.Afisare  
X.Exit

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <ctype.h>

#define NMAX 50

//functie pentru citirea elementelor vectorului
//returneaza numarul de elemente
int citire(int x[])
{
    int nr,i;
    printf("Introduceti numarul de elemente: ");
```

```
scanf("%d", &nr);
for (i=0; i < nr; i++)
{
    printf("Elementul %d = ",i);
    scanf("%d", &x[i]);
}
return nr;
}

//functie pentru afisarea elementelor vectorului
void afisare(int nr, int x[])
{
    int i;
    printf("\nElementele vectorului sunt: ");
    for (i=0; i < nr; i++)
        printf("%d ",x[i]);
    getch();
}

// functia principala
int main(void)
{
    int v[NMAX],n;
    char opt;

    do
    {
        system("cls");
        printf("\nC.Citire");
        printf("\nA.Afisare");
        printf("\nX.Exit");
        printf("\n\nOptiunea dv:");
        opt = toupper(getch());
        printf("\n\n");
        switch(opt)
        {
            case 'C':
                n = citire(v);
                break;
            case 'A': afisare(n, v);
                break;
            case 'X': exit(0);
            default: printf("\nOptiune inexistentă");
                    getch();
                    break;
        }
    } while (1);
    return 0;
}
```

## Probleme

Pentru problemele următoare se vor folosi funcții pentru realizarea fiecărei opțiuni.

1. Să se scrie un program care afișează și prelucrează meniul:
  - 1.Să se calculeze  $n!$ .
  - 2.Să afișeze termenul  $n$  din șirul lui Fibonacci.
  - 3.Să se calculeze puterea unei număr.
  - 4.Să se testeze dacă un număr este prim.
  - 5.Să se calculeze cmmdc pentru două numere.
  
2. Fie un vector cu maxim 50 de elemente întregi. Să se scrie un program care afișează și prelucrează meniul:
  - C.Citire
  - A.Afisare
  - S.Suma elementelor
  - M.Media elementelor
  - E.Cautare element in vector
  - P.Numar de elemente pare
  - N. Numar de elemente divizibile cu 3
  - I.Info autor
  - X.Exit
  
3. Fie o matrice cu maxim 30 de linii și coloane, cu elemente întregi. Să se scrie un program care afișează și prelucrează meniul:
  - C.Citire
  - A.Afisare
  - V.Verificare daca este simetrica
  - F. Căutare element în matrice și afișarea pozițiilor în care se găsește elementul (o poziție = o pereche de coordonate (x, y)).
  - O. Afișare contur matrice.
  - P. Parcurgerea în spirala a matricei se memorează într-un vector, apoi se afișează toate subșirurile de elemente consecutive. Aceasta opțiune va fi lansata în execuție numai dacă matricea este pătratică
  - F. Info autor
  - X. Ieșire

## Funcții recursive

### Exemplu

```
int factorial(int n)
{
    int f;
    if (n==0)
        f = 1;
    else
        f = n*factorial(n-1);
    return f;
}
```

### Probleme

4. Să se scrie programul C care apelează funcția factorial dată ca exemplu.
5. Să se scrie o funcție recursivă care returnează termenul  $n$  din șirul lui Fibonacci.
6. Să se calculeze recursiv  $x^n$ .
7. Să se scrie o funcție care afișează valoarea funcției lui Ackermann:

$$\text{ack}(m,n) = \begin{cases} n + 1, & m = 0 \\ \text{ack}(m-1,1), & n = 0 \\ \text{ack}(m-1, \text{ack}(m,n-1)), & m \neq 0, n \neq 0 \end{cases}$$

8. Să se scrie o funcție recursivă pentru calculul maximului elementelor unui vector.