

7. RECURSIVITATE

A. Probleme rezolvate

1. Pentru un numar natural dat n , sa se calculeze $n!$ folosind o functie recursiva.

Rezolvare

```
#include <stdio.h>
int fact(int n) {
    if (n==0)
        return 1;
    else
        return n*fact(n-1);
}
int main() {
    int n;
    printf("\ndati n: ");
    scanf("%d",&n);
    printf("\nn!=%d", fact(n));
    return 0;
}
```

2. Folosind o functie recursiva, pentru un numar natural n , sa se determine valoarea celui de-al n -lea termen al sirului definit astfel: $a_0=0, a_1=1, a_n=a_{n-1}+a_{n-2}, n>1$;

Rezolvare

```
#include <stdio.h>
int a(int n) {
    if (n==0)
        return 0;
    else if (n==1)
        return 1;
    else
        return a(n-1) + a(n-2);
}
int main() {
    int n;
    printf("\ndati n: ");
    scanf("%d",&n);
    if (n < 0)
        printf("\nIndex negativ");
    else
        printf("\na(n)=%d", a(n));
    return 0;
}
```

B. Probleme propuse

1. Folosind o functie recursiva, pentru un numar natural n , sa se determine valoarea celui de-al n -lea termen al sirului definit astfel:

a) $a_0=0, a_1=1, a_n=(a_{n-1}+a_{n-1})/2, n>1$;

- b) $x_0=1, x_{n+1}=p*x_n, n>0$ si p numar real dat;
 c) $x_0=1, x_{n+1}=a*x_n+b^n, n>0$ si a, b numere reale pozitive;

2. Se considera sirurile (a_n) si (b_n) date prin relatia de recurenta. Sa se determine valorile termenilor a_n si b_n , pentru un numar natural n dat:

- a) $a_0=0, b_0=1, a_{n+1}=(a_n+b_n)/2, b_{n+1}=\sqrt{a_n*b_n}, n>0$;
 b) $a_0=0, b_0=1, a_{n+1}=(a_n+b_n)/2, b_{n+1}=(a_n+2*b_n)/3, n>0$;
 c) $a_0=1, b_0=0, a_{n+1}=(a_n+b_n)/2, b_{n+1}=(7*b_n-a_n)/6, n>0$

Se vor folosi functii recursive pentru definirea sirurilor.

3. Dandu-se doua numere naturale, n si $k, k \leq n$, sa se calculeze C_k^n folosind o functie recursiva.

4. Considerandu-se un sir de numere reale ordonat crescator, x_1, x_2, \dots, x_n , precum si un alt numar real z , sa se determine daca z se afla printre termenii sirului, folosind o functie de cautare binara in sir.

5. Dandu-se un numar natural n , sa se determine daca acesta este palindrom sau nu (un numar este palindrom daca este egal cu numarul format cu cifrele scrise in ordine inversa). Se va folosi o functie recursiva de determinare.

6. Sa se rescrie functia $itoc()$, de la problema 7, laboratorul 6, intr-un mod recursiv.

7. Scrieti un program care citeste de la terminalul standard de intrare un sir de caractere (terminat cu ENTER) si il afiseaza la terminalul standard de iesire in ordinea inversa a caracterelor. Se va folosi o functie recursiva pentru inversarea caracterelor din sirul de caractere.

8. Scrieti o functie recursiva care sa implementeze jocul cunoscut drept „Turnurile din Hanoi”. In acest joc, exista trei tije metalice, precum si un numar de discuri de diferite marimi, care initial sunt toate puse pe tija cea mai din stanga, in ordine crescatoare a diametrelor (discul cel mai mare este la baza tijei). Scopul jocului este acela de a transfera toate discurile pe tija cea mai din dreapta, respectand conditiile urmatoare:

- la fiecare miscare se poate muta un singur disc, pe una dintre cele 3 tije,
- nu se pune niciodata un disc mai mare peste un disc mai mic.

Considerandu-se ca exista n discuri pe tija din stanga, o strategie de joc este urmatoarea:

- a) Se muta primele $n-1$ discuri de pe tija din stanga pe tija din mijloc
 b) Se muta al n -lea disc pe tija din dreapta
 c) Se muta $n-1$ discuri de pe tija din mijloc pe tija din dreapta

9. Sa se scrie o functie recursiva, care sa permita determinarea primelor n polinoame „Legendre” definite astfel:

$$P_0 = 1, P_1 = x,$$

$$P_n = [(2n - 1) / n] * x * P_{n-1} - [(n - 1) / n] * P_{n-2}, n > 1,$$

unde n si x sunt parametri de intrare.