

SEMINAR 3 – PROGRAMAREA CALCULATOARELOR

1. Definiți o structură în care să rețineți datele unei persoane referitoare la CNP, nume, prenume, stare civilă, adresă.
 - a. Citire și afișare date
 - b. Citire și afișare date pentru n persoane
2. Definiți o structură *Produs* care să permită memorarea numelui unui produs, a prețului său unitar și a cantității existente din produsul respectiv într-un depozit. Scrieți o funcție care să ieftinească cu 10% toate produsele aflate într-un depozit al căror stoc depășește un număr real s (stocul unui produs este egal cu produsul dintre preț și cantitate). Informațiile despre produsele din depozitul respectiv sunt memorate într-un tablou t cu n elemente de tip *Produs*.
3. Considerăm în planul xOy dreptunghiuri cu laturile paralele cu axele Ox și Oy . Definiți o structură *Dreptunghi* prin care să puteți reprezenta aceste dreptunghiuri. Scrieți o funcție care primește un tablou unidimensional t de n elemente de tip *Dreptunghi* și le sortează (folosind funcția *qsort*) pe baza ariei lor. Scrieți o funcție care primește un tablou unidimensional t de n elemente de tip *Dreptunghi* și calculează intersecția tuturor dreptunghiurilor din tablou.

(problema intersecției a două cercuri date prin coordonatele punctului centru și raza)

```
//05.11.2024
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct adresa{  
    char localitate[50];  
    char nume_str[50];  
    int nr_str;  
    char bloc[4];  
    int ap;  
};
```

```
struct data{  
    int zi, luna, an;  
};
```

```
struct persoana {  
    char nume[255];  
    struct adresa adr;
```

```

    struct data data_n;
} t[20];

int n;

citire();
afisare();

int main()
{
    citire();
    afisare();
    return 0;
}

void citire(){
    do{
        printf ("\nintroduceti n");
        scanf("%d", &n);
    }while(n<2 || n>=20);
    for (int i = 1; i<=n; i++){
        printf("nume: "); scanf("%s", t[i].nume);
        getchar();
        printf("\n adresa este:\n localitate: "); fgets(t[i].adr.localitate, 50, stdin);
        printf("\nstrada: "); fgets(t[i].adr.num_str, 50, stdin);
        printf("\n nr strada: "); scanf(" %d", &t[i].adr.nr_str);
        getchar();
        printf("\n bloc: "); fgets(t[i].adr.bloc, 4, stdin);
        printf("\n apartament: "); scanf(" %d", &t[i].adr.ap);
        getchar();
        printf("\n data nasterii: ");
        scanf(" %d %d %d", &t[i].data_n.zi, &t[i].data_n.luna, &t[i].data_n.an);
    }
}

void afisare(){
    for (int i = 1; i<=n; i++){
        printf("%s ", t[i].nume);
        printf(" adresa: localitate %s strada %s nr. %d bloc %s ap. %d", t[i].adr.localitate,
t[i].adr.num_str, t[i].adr.nr_str, t[i].adr.bloc, t[i].adr.ap);
        printf(" data nasterii: %d.%d.%d", t[i].data_n.zi, t[i].data_n.luna, t[i].data_n.an);
    }
}

```

```

#include <stdio.h>
#include <stdlib.h>

struct punct{
    float x, y;
};

struct dreptunghi{
    struct punct A, C;
    float arie;
}t[20];

int n;

void citire(){
    printf("Introduceti n: ");
    scanf("%d", &n);
    for(int i = 1; i <= n; i++){
        printf("Coordonatele pentru punctul A: ");
        scanf("%f %f", &t[i].A.x, &t[i].A.y);
        printf("Coordonatele pentru punctul C: ");
        scanf("%f %f", &t[i].C.x, &t[i].C.y);
        t[i].arie = (t[i].C.x-t[i].A.x)*(t[i].C.y-t[i].A.y);
    }
}

void ordonare(){
    int i, j;
    for(i = 1; i<= n-1; i++)
        for(j = i + 1; j<= n; j++){
            if(t[i].arie > t[j].arie){
                t[0] = t[i];
                t[i] = t[j];
                t[j] = t[0];
            }
        }
}

void afisare(){
    int i;
    for(i = 1; i<= n; i++){
        printf("Dreptunghiul %d A.x = %.2f, A.y = %.2f, C.x = %.2f, C.y = %.2f, Aria = %.2f \n", i,
t[i].A.x, t[i].A.y, t[i].C.x, t[i].C.y, t[i].arie);
    }
}

```

```
    }  
}
```

```
int main()  
{  
    citire();  
    afisare();  
    printf("\n");  
    ordonare();  
    afisare();  
  
    return 0;  
}
```

```
//04.11  
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>
```

```
int n;
```

```
struct dreptunghi  
{  
    struct punct {  
        float x,y;  
    } A,C;  
    float Arie;  
} t[21];
```

```
void citire () {  
    scanf ("%d", &n);  
    for (int i=1; i<=n; ++i) {  
        printf ("Coordonatele punctului A sunt:\n");  
        scanf ("%f %f", &t[i].A.x, &t[i].A.y);  
        printf ("Coordonatele punctului C sunt:\n");  
        scanf ("%f %f", &t[i].C.x, &t[i].C.y);  
        t[i].Arie=abs((t[i].C.x-t[i].A.x)*(t[i].C.y-t[i].A.y));  
    }  
}
```

```
void sortare () {  
    for (int i=1; i<n; ++i) {  
        for (int j=i+1; j<=n; ++j) if (t[i].Arie>t[j].Arie) {
```

```

        struct dreptunghi aux;
        aux=t[i];
        t[i]=t[j];
        t[j]=aux;
    }
}

void afisare () {
    for (int i=1; i<=n; ++i) {
        printf ("%0.2f %0.2f), (%0.2f %0.2f); Arie: %0.2f ", t[i].A.x,t[i].A.y,t[i].C.x,t[i].C.y,t[i].Arie);
        printf ("\n");
    }
}

int main()
{
    citire ();
    afisare ();
    if(t[1].A.x<=t[2].A.x && ....)
        printf("da"); else printf("nu");
    //sortare ();
    //afisare ();
    return 0;
}

```

```

//pb.1 / 29.10.2024
#include <stdio.h>
struct date{
    char cnp[14];
    char nume[20];
    char prenume[20];
    char stare_civ[15];
    struct{
        char localitate[15];
        char str[20];
        char bl[5];
        int nr_bl;
        int ap;
    }adresa;
}persoane[10];
int i, n;
int main()

```

```

{ printf("Numarul de persoane:");
  scanf("%d", &n);
  for(i=1;i<=n;i++)
  {
    printf("CNP:");    scanf("%s", persoane[i].cnp);
    printf("Nume:");   scanf("%s", persoane[i].nume);
    printf("Prenume:"); scanf("%s", persoane[i].prenume);
    printf("Stare civila:");scanf("%s", persoane[i].stare_civ);
    printf("Adresa: \n Localitate:"); scanf("%s", persoane[i].adresa.localitate);
    printf("Strada:");   scanf("%s", persoane[i].adresa.str);
    printf("Blocul:");   scanf("%s", persoane[i].adresa.bl);
    printf("Numarul blocului:");    scanf("%d", &persoane[i].adresa.nr_bl);
    printf("Apartament:"); scanf("%d", &persoane[i].adresa.ap);
  }
  for(i=1;i<=n;i++)
  {
    printf("%s %s %s %s %s %s %s %d %d \n", persoane[i].cnp, persoane[i].nume,
    persoane[i].prenume, persoane[i].stare_civ, persoane[i].adresa.localitate, persoane[i].adresa.str,
    persoane[i].adresa.bl, persoane[i].adresa.nr_bl, persoane[i].adresa.ap);
  }
  return 0;
}

```

//Pb 2 / 29.10.2024

```
#include <stdio.h>
```

```
#define N 100
```

```
typedef struct Produs{
  char denumire[30];
  float pretUnitar;
  unsigned int cantitate;
};
```

```
struct Produs v[N];
```

```
void afisare(struct Produs v[], int lungime){
  for(int i=0; i<lungime; i++){
    printf("Produs %d: %s %.2f %d \n", i+1, v[i].denumire, v[i].pretUnitar, v[i].cantitate);
  }
}
```

```
void ieftinire(float s, struct Produs v[], int lungime){
```

```

for(int i=0; i<lungime; i++){
    float stoc=v[i].pretUnitar*v[i].cantitate;
    if(stoc>s){
        v[i].pretUnitar=v[i].pretUnitar-(v[i].pretUnitar*10)/100; //sau v[i].pretUnitar*=0.9;
    }
}
}

```

```

int main()
{
    int n;
    float s;
    printf("n=");
    scanf("%d", &n);
    for(int i=0; i<n; i++){
        printf("Produs #%%d:\ndenumire: ", i+1);    scanf("%s", v[i].denumire);
        printf("Pret unitar: ");    scanf("%f", &v[i].pretUnitar);
        printf("Cantitate: ");    scanf("%d", &v[i].cantitate);
    }

    afisare(v, n);
    printf("s=");    scanf("%f", &s);
    ieftinire(s, v, n);
    afisare(v, n);
    return 0;
}

```

//Pb. 3.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 100

```

```

typedef struct Punct {int x, y;};

```

```

typedef struct Dreptunghi{
    int arie;
    struct Punct a, b;
};

```

```

struct Dreptunghi t[N];

```

```

void ordonare(struct Dreptunghi t[], int n){

```

```

    for (int i = 0; i < n-1; i++)
        for (int j = i+1; j < n; j++)
            if (t[i].arie > t[j].arie)
                {      struct Dreptunghi aux = t[i];    t[i] = t[j];    t[j] = aux;  }
}

```

```

void afisare (struct Dreptunghi t[], int n)
{
    for (int i = 0; i < n; i++)
        printf("%d \n", t[i].arie);
}

```

```

int main()
{
    int n;
    printf("n=");
    scanf("%d", &n);
    for(int i=0; i<n; i++){
        printf("Coordonata x a punctului din stanga sus :\n ");    scanf("%d", &t[i].a.x);
        printf("Coordonata y a punctului din stanga sus :\n ");    scanf("%d", &t[i].a.y);
        printf("Coordonata x a punctului din dreapta jos :\n ");    scanf("%d", &t[i].b.x);
        printf("Coordonata y a punctului din dreapta jos :\n ");    scanf("%d", &t[i].b.y);
        t[i].arie = abs((t[i].b.x - t[i].a.x) * (t[i].b.y - t[i].a.y));
    }
    afisare(t, n);    printf("\n \n ");
    ordonare(t, n);    afisare(t, n);
    return 0;
}

```

// pb. cu fractii:

/*numitor nenul, fractie in forma ireductibila, suma a doua (n) fractii, produsul a doua (n) fractii*/

/*problema cu numere complexe: suma, respectiv produsul a doua (n) numere complexe*/

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
struct fractie{
    int s,j;
}f1,f2;
int cmmdc(int a, int b)
{
    while(a!=b) { if(a>b) { a-=b; } else { b-=a; } }
    return a;
}

```



```

int main()
{
    printf("Fractia 1:");    do{ scanf("%d%d",&f1.s,&f1.j); }while(f1.j==0);
    printf("Fractia 2:");    do{ scanf("%d%d",&f2.s,&f2.j); }while(f2.j==0);
    int d; d=cmmdc(f1.s,f1.j); // printf("%d",d);
    if(d!=1)
    {
        f1.s/=d; f1.j/=d;
    }
    d=cmmdc(f2.s,f2.j);
    if(d!=1)
    {
        f2.s/=d; f2.j/=d;
    }
    system("cls");
    printf("Fractia 1 simplificata: %d / %d\n",f1.s,f1.j);
    printf("Fractia 2 simplificata: %d / %d\n",f2.s,f2.j);
    struct fractie sf; //suma celor doua fractii
    sf.s=f1.s*f2.j+f1.j*f2.s;
    sf.j=f1.j*f2.j;
    printf("%d/%d", sf.s, sf.j);
    return 0;
}

```

//Pb 1/ 28.10.2024

```
#include <stdio.h>
```

```

struct adresa
{
    char localitate[20], strada[20];
    int numar, apartament;
};

```

```

struct persoana
{
    char CNP[13], nume[15], prenm[15], stare_c[10];
    struct adresa a;
};

```

```

int main()
{

```

```

int n;
scanf("%d", &n);
struct persoana p[n]; // vector numit p cu n elemente de tip persoana
for (int i = 0; i < n; i++)
{
    scanf("%s %s %s %s ", p[i].CNP, p[i].nume, p[i].prenm, p[i].stare_c);
    scanf("%s %s %d %d", p[i].a.localitate, p[i].a.strada, &p[i].a.numar, &p[i].a.apartament);
}
for (int i = 0; i < n; i++)
{
    printf("%s %s %s %s ", p[i].CNP, p[i].nume, p[i].prenm, p[i].stare_c);
    printf("%s %s %d %d\n", p[i].a.localitate, p[i].a.strada, p[i].a.numar, p[i].a.apartament);
}
return 0;
}

```

//Pb. 2/ 28.10.2024

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Produs
```

```
{
```

```
    char nume[40];
```

```
    float pret;
```

```
    int cantitate;
```

```
}a[100];
```

```
int n , s;
```

```
void discount(int i )
```

```
{
```

```
    if(a[i].cantitate * a[i].pret > s)
```

```
        a[i].pret -= a[i].pret/10;
```

```
}
```

```
int main()
```

```
{
```

```
    int i ; char null;
```

```
    scanf("%d",&n);
```

```
    getchar();
```

```
    for(i=0;i<n;++i)
```

```
    {
```

```
        fgets(a[i].nume,40, stdin);
```

```
        scanf("%f%d",&a[i].pret,&a[i].cantitate);
```

```

        getchar();
    }
    scanf("%i",&s);
    for(i=0;i<n;++i)
    {
        discount(i);
        printf("%s %f %d\n",a[i].nume,a[i].pret,a[i].cantitate);
    }
    return 0;
}

```

//Pb. 3 / 28.10.2024

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <math.h>
```

```
struct Dreptunghi{
```

```
    struct Punct{
```

```
        float x, y;
```

```
    }A, B;
```

```
    float L, l, arie;
```

```
    }t[100];
```

```
void Afisare(int n, struct Dreptunghi t[])
```

```
{
    for(int i = 0; i < n; i++)
        printf("%f ", t[i].arie);
    printf("\n");
}
```

```
int Intersectari(int n, struct Dreptunghi t[])
```

```
{
    //aici urmeaza!!! Pornim de la a intersecta doua dreptunghiuri!
}
```

```
void Sort(int n, struct Dreptunghi t[])
```

```
{
    int nesortat = 1; // presupun ca vectorul t este nesortat
    while(nesortat)
    {
        nesortat = 0;
        for(int i = 0; i < n-1; i++)
        {
            int j = i + 1;
            if(t[i].arie > t[j].arie)
```

```

        {
            struct Dreptunghi aux;
            aux = t[j];
            t[j] = t[i];
            t[i] = aux;
            nesortat = 1;
        }
    }
}

```

```

int main()
{
    int n;
    scanf("%d", &n);
    for(int i = 0; i < n; i++)
    {
        printf("Coordonate A: ");
        scanf("%f %f", &t[i].A.x, &t[i].A.y);
        printf("Coordonate B: ");
        scanf("%f %f", &t[i].B.x, &t[i].B.y);
        t[i].L = t[i].B.y - t[i].A.y;
        t[i].I = t[i].B.x - t[i].A.x;
        t[i].arie = t[i].L * t[i].I;
        if(t[i].arie < 0)
            abs(t[i].arie);
    }
    Afisare(n, t);
    Sort(n, t);
    Afisare(n, t);

    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
struct Dreptunghi{
    struct Punct{
        float x, y;
    }A, B;
    float L, I, arie;
}

```

```

    }t[100];

void Afisare(int n, struct Dreptunghi t[])
{
    for(int i = 0; i < n; i++)
        printf("%f ", t[i].arie);
    printf("\n");
}

int Intersectari(int n, struct Dreptunghi t[])
{
}

void Sort(int n, struct Dreptunghi t[])
{
    int nesortat = 1; // presupun ca vectorul t este nesortat
    while(nesortat)
    {
        nesortat = 0;
        for(int i = 0; i < n-1; i++)
        {
            int j = i + 1;
            if(t[i].arie > t[j].arie)
            {
                struct Dreptunghi aux;
                aux = t[j];
                t[j] = t[i];
                t[i] = aux;
                nesortat = 1;
            }
        }
    }
}

int main()
{
    int n;
    scanf("%d", &n);
    for(int i = 0; i < n; i++)
    {
        printf("Coordonate A: ");
        scanf("%f %f", &t[i].A.x, &t[i].A.y);
        printf("Coordonate B: ");
    }
}

```

```

scanf("%f %f", &t[i].B.x, &t[i].B.y);
t[i].L = t[i].B.y - t[i].A.y;
t[i].I = t[i].B.x - t[i].A.x;
t[i].arie = t[i].L * t[i].I;
if(t[i].arie < 0)
    abs(t[i].arie);
}
Afisare(n, t);
Sort(n, t);

Afisare(n, t);

return 0;
}

```

Exerciții pentru tipul struct (12 enunțuri)

1. La o școală se ține evidența elevilor la admitere. Astfel, pentru fiecare elev se memorează: numele prenumele, media de admitere, clasa la care este repartizat.

Se cere:

- Să se scrie, în fișierul "admitere.out", pe rânduri disticte, separate prin spațiu, datele despre elevi în ordinea descrescătoare a mediilor.
- Să se scrie câți elevi au media peste 8.

2. O firmă de transporturi rutiere înregistrează pentru fiecare traseu cu plecare din orașul A, următoarele date: numele localității destinație, distanța (în km), timpul necesar pentru parcurgerea acestei distanțe (în minute).

Se cere:

- Să se scrie numele celei mai îndepărtate localități către care firma face transporturi.
- Să se scrie, în fișierul "orașe.out", pe rânduri disticte, separate prin spațiu, toate datele ordonate alfabetic.

3. O asociație de locatari înregistrează pentru fiecare dintre cele n ($n \leq 100$) apartamente ale unui bloc următoarele date: numărul apartamentului, etajul, numărul de persoane, cota de plată la întreținere.

Se cere:

- Să se scrie ce sumă trebuie încasată din plata întreținerii.
- Să se scrie, în fișierul "locatari.out", datele ordonate descrescător după numărul de persoane care locuiesc la adresa respectivă.

4. La un magazin se aduc într-o zi n ($n \leq 50$) produse. Pentru fiecare produs se înregistrează următoarele date: denumire, cantitate și preț (preț unitar, în lei).

Se cere:

- a) Să se afișeze valoarea totală a produselor primite.
- b) Să se scrie, în fișierul "produse.out", pe rânduri distincte, separate prin spațiu, datele despre aceste produse ordonate alfabetic.

5. Pentru fiecare dintre cele n ($n \leq 200$) obiecte ale unei colecții se înregistrează următoarele date: denumirea, anul achiziției și valoarea (în lei).

Se cere:

- a) Să se afișeze valoarea totală a obiectelor achiziționate în intervalul 2005-2020.
- b) Să se scrie, în fișierul "colectie.txt", pe rânduri distincte, separate prin spațiu, datele despre obiectele colecției ordonate descrescător după valoare.

6. Se citesc, din fișierul "ploi.in", de pe prima linie un număr natural n ($n \leq 20$), iar de pe următoarele $3 \cdot n$ linii, numere ce reprezintă, în această ordine: ziua, luna și cantitatea de precipitații înregistrată în acea zi.

Se cere:

- a) Să se afișeze lista informațiilor ordonate în funcție de cantitatea de precipitații, descrescător.
- b) Să se scrie media precipitațiilor înregistrate în fișierul dat.

7. Se citește de la tastatură un număr natural n ($n \leq 50$), apoi se citesc următoarele date pentru fiecare dintre cele n titluri de ziare primite la un chioșc: denumire, preț (preț unitar, în lei), număr de exemplare.

Se cere:

- a) Să se scrie, în fișierul "date.out", valoarea totală a ziarelor primite.
- b) Să se afișeze toate datele ordonate alfabetic după denumirea ziarelor (pentru fiecare ziar câte un rând, datele fiind separate prin spațiu).

8. Pentru o clasă de elevi se pregătesc manualele, în număr de n titluri ($n \leq 12$). Pentru fiecare manual se înregistrează: denumirea (disciplina), editura, anul apariției și prețul unitar (în lei).

Se cere:

- a) Să se scrie, în fișierul "manuale.out", pe rânduri distincte, separate prin spațiu, toate datele despre manuale, ordonate alfabetic după denumire.
- b) Să se afișeze valoarea totală a manualelor care vor reveni unui elev.

9. Se citesc de la tastatură date despre n angajați ($n \leq 50$): nume, prenume, funcție, vechime (în ani).

Se cere:

- a) Să se afișeze câte persoane au funcția de "muncitor".
- b) Să se scrie, în fișierul "date.out", pe n rânduri distincte, separate prin spațiu, toate datele celor n angajați, în ordine crescătoare a vechimii.

10. Pentru elevii unei clase (cel mult 30) se înregistrează următoarele date: nume, prenume, medie, număr de absențe motivate, număr de absențe nemotivate.

Se cere:

- a) Să se afișeze numărul total de absențe nemotivate.
- b) Să se scrie, în fișierul "bursieri.out", pe rânduri distincte, separate prin spațiu, numele, prenumele și media elevilor care au mai puțin de 10 absențe nemotivate și media cel puțin 9.50.

11. Pentru elevii unei clase (cel mult 25) se înregistrează următoarele date: nume, prenume, medie, număr de absențe motivate, număr de absențe nemotivate.

Se cere:

- a) Să se afișeze, pe un rând, separate prin spațiu, cea mai mare medie, respectiv cea mai mică medie.
- b) Să se scrie, în fișierul "absențe.out", pe rânduri distincte, separate prin spațiu, numele, prenumele și media elevilor care au mai multe absențe nemotivate decât cele motivate.

12. La o bibliotecă, pentru fiecare carte se înregistrează: autorul (nume și prenume), titlul, editura, anul apariției, prețul.

Se cere:

- a) Să se scrie, în fișierul "biblioteca.out", pe rânduri distincte, separate prin spațiu, datele despre cărțile apărute după 2010.
- b) Să se afișeze valoarea totală a cărților din bibliotecă.