

Laborator 12 – Probleme recapitulative

1. Probleme cu șiruri

1.1 Să se scrie un program care să creeze un șir cu n elemente, care au proprietatea că $elem_{i+1} = elem_i + 3$, ținând cont că n și primul element din șir se citește de la tastatură.

```
#include<stdio.h>
#define MAX 100
int main()
{
int i,j,k,n;
double a[MAX];
printf("Introduceti numarul de elemente ale sirului n=");
scanf("%d",&n);
printf("Introduceti primul element din sir:\n");
printf("a[0]=");
scanf("%lf",&a[0]);
printf("Elementele sirului sunt: \n%.2lf ",a[0]);
for(i=1;i<n;i++){
    a[i]=a[i-1]+3;
    printf("%.2lf ",a[i]);
}
printf("\n");
}
```

1.2 Se citește un șir de n numere de la tastatură. Să se scrie un program care elimină din șir elementele care nu aparțin șirului lui Fibonacci (elementele șirului lui Fibonacci au proprietatea: $elem_{i+1} = elem_i + elem_{i-1}$).

```
#include <stdio.h>
#define MAX 100
int ApartineSirFibonacci(int nr);
void main()
{
int sir[MAX];
int i,n,j;
printf("Introduceti numarul de elemente ale sirului n=");
scanf("%d",&n);
for (i = 0; i < n; i++){
    printf("sir[%d]= ", i);
    scanf("%d", &sir[i]);
}
for(i=0;i<n;i++){
    if(ApartineSirFibonacci(sir[i])==0){
        // Eliminare element din sir
        for (j = i; j < n - 1; j++)
        {
            sir[j] = sir[j + 1];
        }
    }
}
```

```
    }
    n--;
  }
}
printf("Numerele care apartin sirului lui Fibonacci sunt:\n");
for (i = 0; i < n; i++){
    printf("%d ", sir[i]);
}
printf("\n");
}

// returneaza 1 daca nr. apartine sirului lui Fibonacci, 0 daca nu apartine
int ApartineSirFibonacci(int nr)
{
// Sirul lui fibonacci: 1, 1, 2, 3, 5, 8, 13, ....
int a=1;
int b=1;
int temp;
if ((nr == 0)||(nr==1))
    return 1;

while (b < nr){
    temp = a + b;
    a = b;
    b = temp;
}

if (b == nr)
    return 1;

return 0;
}
```

Programul conține o funcție: **ApartineSirFibonacci()** care verifică dacă un element al șirului introdus aparține șirului lui Fibonacci. Acest lucru se realizează prin generarea elementelor șirului lui Fibonacci până la elementul care se verifică și care este parametru de intrare în funcție și verificarea dacă ultimul element al șirului lui Fibonacci generat este egal cu elementul care se verifică. Dacă rezultatul este pozitiv, funcția returnează valoarea 1, altfel 0. Dacă funcția returnează 0, elementul care se verifică este eliminat din șir. La final se afișează elementele care rămân.

1.3 Să se scrie un program care citește n cuvinte de la tastatură și le ordonează alfabetic crescător.

```
#include<stdio.h>
#include<string.h>
void main()
{
    char cuvinte[10][25],aux[25];
    int n,i,gasit;
```

```
printf("Numarul de cuvinte, n=");
scanf("%d",&n);
printf("Introduceti cuvintele:\n");
for(i=0;i<n;i++)
{
    printf("cuvant: ");
    scanf("%s",&cuvinte[i]);
}
do
{
    gasit=0;
    for(i=0;i<n-1;i++)
        if(strcmp(cuvinte[i],cuvinte[i+1])>0)
        {
            strcpy(aux,cuvinte[i]);
            strcpy(cuvinte[i],cuvinte[i+1]);
            strcpy(cuvinte[i+1],aux);
            gasit=1;
        }
    while(gasit);
    printf("Cuvintele ordonate alfabetic crescator sunt:\n");
    for(i=0;i<n;i++)
        printf("%s ",cuvinte[i]);
    printf("\n");
}
```

2. Probleme cu matrice

2.1. Se citește de la tastatura o matrice pătratică. Să se scrie un program prin care să se formeze un șir care să conțină elementele cu suma indicilor pară.

```
#include<stdio.h>
#include<math.h>
#define MAX 20
void main(){
double a[MAX][MAX],b[MAX];
int i,j,n,k=0;
printf("Introduceti ordinul matricei n=");
scanf("%d", &n);
printf("Introduceti elementele matricei: \n");
for(i = 0; i < n; i++){
    for(j = 0; j < n; j++){
        printf("a[%d][%d]=",i,j);
        scanf("%lf", &a[i][j]);
        if((i+j)%2==0){
            b[k]=a[i][j];
            k++;
        }
    }
}
```

```
printf("Elementele sirului b sunt:\n");
for(i=0;i<k;i++)
printf("b[%d]=%.2lf ",i,b[i]);
printf("\n");}
```

2.2. Se consideră o matrice pătratică. Să se scrie un program care determină poziția elementului maxim al matricei și produsul scalar al liniei și coloanei la intersecția cărora se află elementul minim.

```
#include<stdio.h>
#define MAX 20
void main(){
float mat[MAX][MAX],prod=0,vmin,vmax;
int i,j,n,imin=0,jmin=0,imax=0,jmax=0;
printf("Introduceți ordinul matricei:");
scanf("%d", &n);
printf("Introduceți elementele matricei: \n");
for(i = 0; i < n; i++){
    for(j = 0; j < n; j++){
        printf("mat[%d][%d]=",i,j);
        scanf("%f", &mat[i][j]);
    }
}
vmax=mat[0][0];
vmin=mat[0][0];
for(i=0;i<n;i++){
    for(j=1;j<n;j++){
        if(mat[i][j]>vmax){
            imax=i;
            jmax=j;
        }
        if(mat[i][j]<vmin){
            vmin=mat[i][j];
            imin=i;
            jmin=j;
        }
    }
}
printf("Elementul maxim se afla pe pozitia %d,%d\n",imax,jmax);
i=0;
for(j=0;j<n;j++){
    if(i!=imin||j!=jmin){
        prod=prod+mat[imin][j]*mat[i][jmin];
        i++;
    }
}
printf("Produsul scalar al liniei si coloanei la intersecția carora se afla elementul minim este:
%f\n",prod);
}
```

În program se determină mai întâi elementele maxim și minim precum și pozițiile acestora în matrice, iar apoi se calculează produsul scalar al elementelor de pe linia și coloana la intersecția cărora se află elementul minim. Acest lucru este realizat utilizând o structură iterativă condusă de variabila j, variabila i incrementându-se la fiecare pas. Pentru fiecare nouă iterație se verifică dacă variabila j nu se află pe poziția jmin, caz în care nu se calculează produsul $mat[imin][j]*mat[i][jmin]$ (la $j=jmin$ și i are valoarea imin).

2.3 Fie o matrice pătratică. Să se scrie un program care determină poziția elementelor, care au următoarea proprietate: sunt cele mai mici pe linia pe care se găsesc și cele mai mari pe coloană (dacă astfel de elemente există).

```
#include<stdio.h>
#define MAX 20
int maxim(float mat[][MAX], float vmin, int jmin, int n);
void main()
float mat[MAX][MAX],vmin;
int i,j,n,jmin=0,ind=0,imin=0;
printf("Introduceti ordinul matricei:");
scanf("%d", &n);
printf("Introduceti elementele matricei: \n");
for(i = 0; i < n; i++){
    for(j = 0; j < n; j++){
        printf("mat[%d][%d]=",i,j);
        scanf("%f", &mat[i][j]);
    }
}
for(i=0;i<n;i++){
    vmin=mat[i][0];
    for(j=1;j<n;j++){
        if(mat[i][j]<vmin){
            vmin=mat[i][j];
            jmin=j;
            imin=i;
        }
    }
    if(maxim(mat,vmin,jmin,n)){
        ind=1;
        printf("Elementul de pe pozitia i=%d, j=%d, avand valoarea
%.2f are proprietatea ceruta\n",imin,jmin,mat[imin][jmin]);
    }
}
if(ind==0)
    printf("Nu exista elemente care sa satisfaca proprietatea ceruta\n");
}

int maxim(float mat[][MAX], float vmin, int jmin, int n)
{
    int i;
    for(i=0;i<n;i++)
        if(vmin>=mat[i][jmin])
```

```

        return 1;
    else
        return 0;
}

```

Programul determină valoarea minimă de pe fiecare linie a matricei introduse, iar pentru fiecare minim găsit se apelează funcția maxim() care returnează 1 dacă minimum găsit este maxim pe coloana sa sau 0 în caz contrar. Se folosește variabila ind care indică dacă nu s-au găsit elemente care au această proprietate în matricea introdusă.

2.4. Se consideră o matrice cu m linii și n coloane, având elemente reale. Să se scrie un program care calculează următoarea expresie:

$$E = \sqrt[m]{\prod_{i=1}^m \left(\sum_{j=1}^n a_{i,j}^2 \right)}$$

```

#include<stdio.h>
#include<math.h>
#define MAX 20
void main(){
double a[MAX][MAX],suma,prod=1;
int i,j,m,n;
printf("Introduceti numarul de linii ale matricei m(>1)=");
scanf("%d", &m);
while(m==0){
    printf("Reintroduceti valoarea lui m=");
    scanf("%d",&m);
}
printf("Introduceti numarul de coloane ale matricei n=");
scanf("%d", &n);
printf("Introduceti elementele matricei: \n");
for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        printf("a[%d][%d]=",i,j);
        scanf("%lf", &a[i][j]);
    }
}
for(i=0;i<m;i++){
    suma=0;
    for(j=0;j<n;j++)
        suma=suma+pow(a[i][j],2);
    prod=prod*suma;
}
printf("E=%lf\n",pow(prod,1.0/(float)m));}

```

În program, se calculează pentru fiecare linie a matricei suma pătratelor elementelor de pe fiecare linie (variabila suma), iar apoi se înmulțește valoarea acesteia cu variabila prod inițializată cu 1. Variabila suma se reinițializează cu 0 după ce se trece la o linie nouă.