

- Să se scrie funcția pentru citirea unui vector de la tastatură.

Funcția nu are parametri de intrare. Parametrii de ieșire sînt vectorul și numărul de elemente, pentru care se simulează transferul prin adresă.

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

void citire(int v[],int* n)
{ int i,er, p;
  printf("\nn=");scanf("%d",n);
  for(i=0;i<*n;i++)
  {printf("v(%d)=",i);
   do {p=scanf("%d",&v[i]);}
   while(p!=1);}
}

void main()
{
    int n, v[50], i;
    printf("Introduceți elementele vectorului");
    citire(v, &n);
    printf("Vectorul este");
    for(i=0; i<n; i++)
        printf("v[%d]=%d\t", i, v[i]);
    _getch();
}
```

- Să se scrie funcția pentru afișarea unui vector pe ecran.

Funcția are ca parametri de intrare vectorul și numărul de elemente.

```
void afisare(float v[],int n)
{ int i;
  printf("\n"); for(i=0;i<n;i++) printf("\t%5.2f",v[i]);}
```

- Să se scrie funcția pentru găsirea elementului minim dintr-un vector.

Funcția are ca parametri vectorul și numărul de elemente și returnează, prin numele ei, elementul minim.

```
float minim(float v[],int n)
{ float m; int i;
  m=v[0];
  for(i=0;i<n;i++)
      if(m>v[i])m=v[i];
  return(m);}
```

- Să se scrie funcția pentru găsirea elementului minim și a primei poziții de apariție a acestuia într-un vector.

Funcția are ca parametri vectorul, numărul de elemente și adresa unde se va reține prima poziție de apariție a minimului. Prin numele funcției se returnează valoarea minimului.

```
float minim(float v[],int n,int *poz)
{ float m;
  m=v[0];
  *poz=0;
  int i;
  for(i=0;i<n;i++)
      if(m>v[i]){m=v[i];
                *poz=i;}
  return(m);}
```

- Să se scrie funcția pentru găsirea elementului minim și a ultimei poziții de apariție a acestuia într-un vector.

Funcția are ca parametri vectorul, numărul de elemente și adresa unde se va reține ultima poziție de apariție a minimului. Prin numele funcției se returnează valoarea minimului.

```
float minim(float v[],int n,int *poz)
{ float m;
  m=v[0];
  *poz=0;
  int i;
  for(i=0;i<n;i++)
    if(m>=v[i]){m=v[i];
               *poz=i;}
  return(m);}
```

- Să se scrie funcția pentru găsirea elementului maxim și a tuturor pozițiilor sale de apariție într-un vector.

Funcția are ca parametri vectorul, numărul de elemente, vectorul unde se vor reține pozițiile maximului și adresa unde se va scrie numărul de apariții ale maximului.

```
float minim(float v[],int n,int poz[],int *nrpoz)
{ float m;
  int i;
  m=v[0]; poz[0]=0;
  *nrpoz=1;
  for(i=1;i<n;i++)
    if(m>v[i]) {m=v[i];
               poz[0]=i;
               *nrpoz=1;}
    else if(m==v[i]) {poz[*nrpoz]=i;
                     (*nrpoz)++;}
  return(m);}
```

- Să se scrie funcția pentru inserarea unui 0 între fiecare două elemente ale unui vector.

Funcția are ca parametri vectorul și adresa numărului de elemente ale sale. La adresa respectivă se va reține noul număr de elemente rezultat în urma prelucrării. Nu se obține nici un rezultat prin numele funcției.

```
void inserare(float v[],int* n)
{ int i,j,k;
  k=*n;
  for(i=0;i<k-1;i++)
    {for(j=*n;j>2*i+1;j--) v[j]=v[j-1];
     v[2*i+1]=0;
     (*n)++;}
}
```

- Să se scrie funcția pentru crearea unui vector din elementele unui vector dat, inserând câte un 0 între fiecare 2 elemente ale acestuia.

Funcția are ca parametri vectorul inițial și numărul său de elemente, vectorul rezultat și adresa unde se va scrie numărul de elemente ale vectorului rezultat. Nu se întoarce nici o valoare prin numele funcției.

```
void inserare(float v[],int n,float v1[],int* n1)
{ int i;
  *n1=0;
  for(i=0;i<n-1;i++)
    {v1[2*i]=v[i];
     v1[2*i+1]=0;
     (*n1)+=2;}
  v1[*n1]=v[n-1];
  (*n1)++;}
```