

## VECTORI (I)

### 1. SCOPUL LUCRĂRII

În această lucrare se vor studia tablouri unidimensionale (vectori).

### 2. BREVIAR TEORETIC

Un tablou reprezintă o colecție de date de același tip. Tablourile unidimensionale sunt denumite și vectori.

La declararea unui tablou se specifică numărul de elemente ale fiecărei dimensiuni, incluzând fiecare dintre aceste numere între paranteze drepte. Indexul inferior al fiecărei dimensiuni este 0.

Sintaxa pentru declararea unui vector este următoarea:

*tipul\_datelor nume\_vector[Numărul\_total\_de\_elemente];*

Exemplu: tab -vector ce conține 100 de numere reale:

*double tab[100];*

Elementele tabloului se accesează indexând corespunzător numele tabloului.

Exemplu:

*tab[10]=tab[10]+1;*

Un vector poate fi inițializat încă din faza de declarare, ca în exemplul următor:

*int x[4]={-1, 2, 0, 56};*

### 3. DESFĂȘURAREA LUCRĂRII

Se vor edita și apoi executa programele descrise în continuare.

#### **Programul nr. 1**

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Să se calculeze maximumul din vector.

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h> { #include<iostream>
#define N 7      } using namespace std;
int void main(void)
{
    int a[N];
    int i;
clrscr();
    for(i=0;i<N;i++){
```

```

    printf("a[%d]=",i); scanf("%d",&a[i]);    cout<<"a["<<i<<"]= ";
    scanf("%d",&a[i]);    cin>>a[i];
    int max=a[0];
    for(i=1;i<N;i++)
        if(a[i]>max)max=a[i];
    printf("maximul = %d",max);    cout<<"maximul = "<<max;
getch();
}

```

### Programul nr. 2

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Să se calculeze cate numere pare sunt în vector.

#### Sursa programului:

```

#include <stdio.h>
#include <conio.h>
#define N 7
void main(void)
{
    int a[N];
    int i;
clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
        scanf("%d",&a[i]);
        int contor=0;
        for(i=0;i<N;i++)
            if(a[i]%2==0)contor++;
        printf("Numarul de numere pare=%d",contor);
getch();
    }
}

```

### Programul nr. 3

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Să se memoreze toate numerele pare din vectorul a într-un vector b. Sa se afișeze vectorul b.

#### Sursa programului:

```

#include <stdio.h>
#include <conio.h>
#define N 7
void main(void)

```

---

```
{
    int a[N],b[N];
    int i;
    clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
        scanf("%d",&a[i]);}
    int j=0;//index in vector b
    for(i=0;i<N;i++)
        if(a[i]%2==0){
            b[j]=a[i];
            j++; }
    //afisare vector b:
    if(j==0)printf("Nu sunt numere pare in vectorul a");
    else
        for(i=0;i<j;i++)
            printf("%d\n",b[i]);
    getch();
}
```

**Programul nr. 4**

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Să se afișeze dacă toate numerele introduse sunt egale între ele.

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
#define N 7
void main(void)
{
    int a[N];
    int i;
    clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
        scanf("%d",&a[i]);}
    int suntEgale=1; //presupunem ca sunt toate egale
    //Le comparăm pe toate cu primul element:
    for(i=1;i<N;i++)
        if(a[i]!=a[0]){
```

---

```

    suntEgale=0;
    break; }
if(suntEgale)printf("Sunt toate egale");
else printf("Nu sunt toate egale");
getch();
}

```

### Programul nr. 5

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Să se afișeze dacă toate numerele introduse sunt diferite între ele.

#### Sursa programului:

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define N 7
int void main(void)
{
    int a[N];
    int i,j;
clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
        scanf("%d",&a[i]);}
    //Se parcurge vectorul
    // se compara elementul curent cu toate elementele de dupa el
    for(i=0;i<N-1;i++)
        for(j=i+1;j<N;j++)
            if(a[i]==a[j]){
                printf("Nu sunt toate diferite");
                getch();
                exit(0);}
    printf("Sunt toate diferite");
getch();
}

```

### Programul nr. 6

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Se citește un număr întreg x. Să se afișeze dacă numărul x este prezent în vector sau nu (algoritmul de căutare liniară).

---

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
#define N 7
void main(void)
{
    int a[N];
    int i;
    clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
        scanf("%d",&a[i]);}
    int x;
    printf("x="); scanf("%d",&x);
    int estePrezent=0; //presupunem ca nu este
    for(i=0;i<N;i++)
        if(a[i]==x){
            estePrezent=1;
            break; }
    if(estePrezent)printf("%d este prezent in vector",x);
    else printf("%d nu este prezent in vector",x);
    getch();
}
```

**Programul nr. 7**

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Se citește un număr întreg *poz*, ce reprezintă o poziție în vector. Să se șteargă din vector numărul aflat pe poziția *poz*, și să se afișeze noul vector.

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
#define N 7
void main(void)
{
    int a[N];
    int i;
    clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
```

```

    scanf("%d",&a[i]);}
    int poz;
    printf("poz="); scanf("%d",&poz);
    //Parcurgem vectorul de la indexul poz+1 pana la sfarsit,
    // si mutam prin copiere, elementul curent spre stanga cu o
    //pozitie
    for(i=poz+1;i<N;i++)
        a[i-1]=a[i];
    //Afisare:
    for(i=0;i<N-1;i++)
        printf("%d\n",a[i]);
    getch();
}

```

### **Programul nr. 8**

Se citesc N numere întregi de la tastatură și se memorează într-un vector a. Se citește un număr întreg *poz*, ce reprezintă o poziție în vector. Se citește un număr întreg x. Să se insereze numărul x pe poziția *poz*, în vectorul a și să se afișeze noul vector.

#### **Sursa programului:**

```

#include <stdio.h>
#include <conio.h>
#define N 7
void main(void)
{
    int a[N]; a[N+1];
    int i;
    clrscr();
    for(i=0;i<N;i++){
        printf("a[%d]=",i);
        scanf("%d",&a[i]);}
    int poz;
    printf("poz="); scanf("%d",&poz);
    int x;
    printf("x="); scanf("%d",&x);

```

*/\* Mutam incepand de la pozitia poz, spre dreapta cu o pozitie, toate elementele vectorului. Aceasta mutare o facem parcurgand vectorul in sens invers, pentru a nu distruge informatia din vector \*/*

---

```
for(i=N-1;i>=poz;i--)
    a[i+1]=a[i];
// Inseram pe x:
a[poz]=x;
// Afisare:
for(i=0;i<=N;i++)
    printf("%d\n",a[i]);
getch();
}
```

**Programul nr. 9**

Se citesc de la tastatură două mulțimi de numere întregi,  $a$  de 20 de elemente și  $b$  de 10 de elemente (elementele, în fiecare mulțime, se presupun distincte). Se calculează și se afișează reuniunea celor două mulțimi de numere.

Vom folosi constantele simbolice  $na$  și  $nb$ , pentru stocarea dimensiunii mulțimilor  $a$  și  $b$ .

**Sursa programului:**

```
#include <stdio.h>
#include <conio.h>
#define na 20
#define nb 10
void main(void)
{
    int a[na], b[nb];
    //vectorul folosit pentru reprezentarea multimii reuniune:
    int c[na+nb]; //dimensiune acoperitoare
    int i, j, k;
    int esteComun; //variabila semafor
    clrscr();
    //citire vector a:
    for(i=0; i<na; i++){
        printf("a[%d]= ", i);
        scanf("%d", &a[i]);
    }
    //citire vector b:
    for(i=0; i<nb; i++){
        printf("b[%d]= ", i);
        scanf("%d", &b[i]);
    }
}
```

---

```

/* Vectorul c consta din vectorul a si elementele din b care nu sunt si
in a. */
//Copiem in c pe a:
for(i=0;i<na;i++)
    c[i]=a[i];
//Adaugam in c si elementele din b care nu sunt si in a:
k=na-1;//indexul ultimei componente adaugate in c
for(i=0;i<nb;i++){
    //este prezent b[i] in a?
    esteComun=0;//initializare: nu este prezent.
    for(j=0;j<na;j++)
        if(b[i]==a[j]){
            esteComun=1;//este prezent
            break;//din for(j)
        }
    //end if
    if(esteComun==0){//se adauga b[i] in multimea c
        k++;//cresc index curent in multimea c
        c[k]=b[i]; }
    }
}
//for(i)
//Afisarea multimii reuniune:
for(i=0;i<=k;i++)
    printf("\n %d",c[i]);
getch();
}

```

#### 4. PROBLEME PROPUSE

1. Se citește un număr natural de tip long int. Să se copieze toate cifrele într-un vector.
  2. Se citește un număr natural N. Să se copieze primele N numere prime într-un vector.
  3. Să se realizeze un program care rotește un vector spre stânga cu o poziție.
  4. Să se realizeze un program care rotește un vector spre dreapta cu o poziție.
  5. Sa se calculeze intersecția a doua mulțimi de numere întregi, reprezentate prin vectori.
  6. Să se realizeze un program care analizează daca într-un vector elementele sunt în progresie aritmetică.
-