

Programare procedurală

– Laborator 4 –

I. Tablouri unidimensionale (vectori)

Declarare:

```
<tip_date> <nume_vector>[<dimensiune>];
```

Observații:

- ```
int main()
{
 int v[]; //eroare

}
```
- ```
int main()
{
    int v[]={7,2,3};   /* compilatorul determină dimensiunea: 3
                        => se alocă 3x4 bytes */
    printf("%d\n",v[0]); //se afișează 7
    scanf("%d",&v[3]);   /* se suprascrive o zonă de memorie ce
                        poate conține informații
                        importante(!) */

    printf("%d\n",v[3]);
    .....
}
```
- ```
int main()
{
 int n;
 scanf("%d",&n);
 int v[n]; // e ok;

}
```
- ```
#define MAX 100
int main()
{
    int v[MAX];        // e ok;
    .....
}
```
- ```
int v[100]; float w[100];
int main()
{
 printf("%d %f",v[7],w[15]); /* se va afișa 0 - variabilă
 globală = inițializare automată cu 0 */

}
```
- Un vector declarat **local**, cu elemente neinițializate poate memora **orice** valoare (de tipul de date corespunzător).
- Principala cauză a erorilor: depășirea limitelor vectorului.  
*Sfat:* verificați întotdeauna că indicii folosiți pentru accesarea elementelor se încadrează între marginile vectorului (inferioară și superioară).

## II. Tablouri multidimensionale (matrice)

<tip\_elemente> <nume\_tablou>[<dim\_1>][<dim\_2>]...[<dim\_n>;

### Observații:

- Inițializare la declarare:  

```
int patrat[4][2] = {{1,1},{2,4},{3,9},{4,16}};
```

  

```
int patrat[4][2] = {
1,1,
2,4,
3,9,
4,16,
};
```

  

```
int patrat[][2] = {1,1, 2,4, 3,9, 4,16};
```

### Probleme

1. Se citesc numerele naturale  $n$ ,  $m$  și apoi două mulțimi A și B cu  $n$ , respectiv  $m$  numere întregi cuprinse între  $[-x, x]$ ,  $x \leq 2000$ . Să se afișeze numărul de elemente comune mulțimii.  
(Indicație: mulțimile A și B nu se vor memora - se va crea un vector de frecvență).
2. Se citesc:  $n$ , cele  $n$  elemente ale unui vector sortat crescător, apoi  $x$  și  $y$  două elemente din vector. Să se afișeze toate elementele vectorului cuprinse între  $x$  și  $y$ .  
*Optim:* folosiți căutarea binară. *Observație:* <https://research.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html>
3. Se citesc  $x$ ,  $y$ , două numere mari (fiecare având peste 20 de cifre). Să se calculeze suma lor (folosind vectori). Tratați cazurile următoare:
  - a. numerele sunt naturale.
  - b. numerele sunt întregi.
4. Se citesc de la tastatură  $m$  și  $n$  numere naturale nenule reprezentând dimensiunile unei matrice și elementele matricei. Să se construiască și să se afișeze matricea transpusă.  
$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$
$$A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$
5. Să se parcurgă o matrice în spirală.  
*Exemplu:* Pentru  $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$   
se va afișa 1, 2, 3, 6, 9, 12, 11, 10, 7, 4, 5, 8.
6. Să se creeze o matrice patratică, în spirală, după regulile:
  - o numerele pornesc de la 1, din 1 în 1, în ordine crescătoare;
  - o după fiecare număr neprim  $x$  se adaugă cel mai mic divizor propriu al său, după care se continuă cu  $x+1$

*Exemplu:*

|    |    |    |    |   |
|----|----|----|----|---|
| 1  | 2  | 3  | 4  | 2 |
| 11 | 12 | 2  | 13 | 5 |
| 2  | 16 | 2  | 14 | 6 |
| 10 | 3  | 15 | 2  | 2 |
| 3  | 9  | 2  | 8  | 7 |

7. Se citește o matrice  $A$  de dimensiuni  $N \times N$  ( $1 \leq N \leq 100$ ), ( $0 \leq A[i][j] < 2^{32}$ ). Să se efectueze o rotire spre dreapta a matricei  $A$ .

*Exemplu:*

|   |   |    |   |   |
|---|---|----|---|---|
| 1 | 2 |    | 3 | 1 |
| 3 | 4 | => | 4 | 2 |

8. Se citește o matrice  $A$  de dimensiuni  $N \times N$  ( $1 \leq N \leq 16$ ), ( $0 \leq A[i][j] < 16$ ). Să se calculeze determinatul matricei  $A$ .

9. Pătratul magic: <http://campion.edu.ro/arhiva/index.php?page=problem&action=view&id=103>

10. Pentru cei care se plictisesc: <http://www.infoarena.ro/problema/mayonaka> (soluție de 100p).