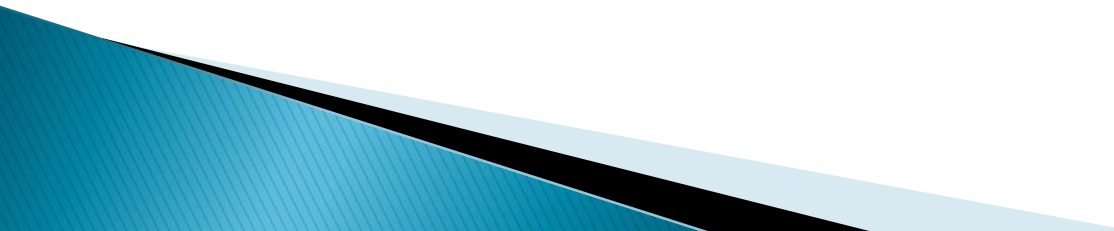


# S02 – SDD

Alin Zamfiroiu

[alin.zamfiroiu@csie.ase.ro](mailto:alin.zamfiroiu@csie.ase.ro)

# S02 – Continut

- ▶ Articol televizor
  - ▶ Citire articol
  - ▶ Citire vector
  - ▶ Citire matrice
  - ▶ Matrice in zig zag
  - ▶ Tablou n-dimensional
  - ▶ Sortare vector
- 

# S02 – Articol televizor

```
struct televizor  
{  
    char*marca;  
    int diagonala;  
};
```



# S02 – Citire articol

```
void main()
{
    televizor t;
    t.marca=(char*)malloc((strlen("Anonim")+1)*sizeof(char));
    strcpy(t.marca,"Anonim");
    t.diagonala=10;
    citireArticol(t);
    printf("%s %d",t.marca,t.diagonala);
}
```



```
void citireArticol(televizor t)
{
    char aux[20];
    printf("Introduceti marca televizorului:");
    scanf("%s",&aux);
    t.marca=(char*)malloc((strlen(aux)+1)*sizeof(char));
    strcpy(t.marca,aux);
    printf("Introduceti diagonala televizorului:");
    scanf("%d", &t.diagonala);
}
```

# S02 – Citire articol

```
void citireArticol(televizor* t)
{
    char aux[20];
    printf("Introduceti marca televizorului:");
    scanf("%s",&aux);
    t->marca=(char*)malloc((strlen(aux)+1)*sizeof(char));
    strcpy(t->marca,aux);
    printf("Introduceti diagonala televizorului:");
    scanf("%d", &t->diagonala);
}
```

```
televizor t;
citireArticol(&t);
```

**If you think you can  
make it better...**

**Do it.**

# S02 – Citire articol

```
void citireArticol(televizor& t)
{
    char aux[20];
    printf("Introduceti marca televizorului:");
    scanf("%s",&aux);
    t.marca=(char*)malloc((strlen(aux)+1)*sizeof(char));
    strcpy(t.marca,aux);
    printf("Introduceti diagonala televizorului:");
    scanf("%d", &t.diagonala);
}
```

```
void main()
{
    televizor t;
    citireArticol(t);
    printf("%s %d",t.marca,t.diagonala);
}
```

# S02 – Citire vector

```
televizor* vector;  
int lungime;  
citireVector(&vector,&lungime);  
afisareVector(vector,lungime);
```



# S02 – Citire vector

```
void citireVector(televizor**v,int*n)
{
    printf("Introduceti numarul de televizoare");
    scanf("%d",n);
    (*v)=(televizor*)malloc(sizeof(televizor)*(*n));
    for(int i=0;i<*n;i++)
    {
        printf("Introduceti marca televizorului:");
        char aux[20];
        scanf("%s",&aux);
        ((*v)[i]).marca=(char*)malloc(sizeof(char)*(strlen(aux)+1));
        strcpy((*v)[i].marca,aux);
        printf("Introduceti Diagonala televizorului:");
        scanf("%d",&(*v)[i].diagonala);
    }
}
```

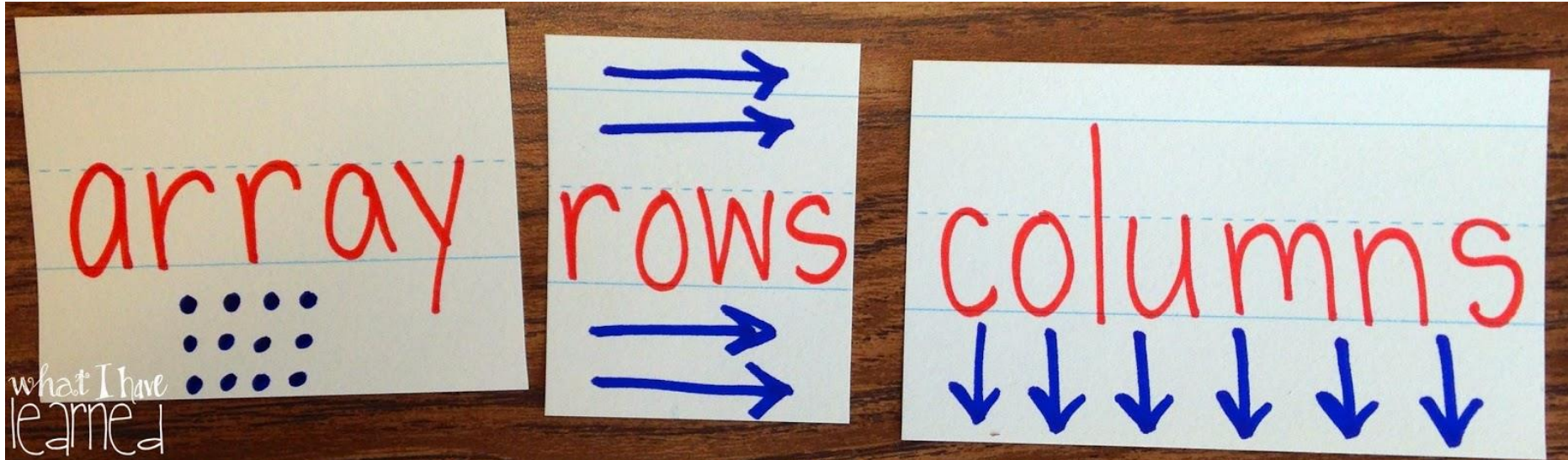
```
void afisareVector(televizor*vector,int n)
{
    for(int i=0;i<n;i++)
        printf("%s %d\n",vector[i].marca,vector[i].diagonala);
}
```



# S02 – Citire vector

```
void citireVector2(televizor* &v,int &n)
{
    printf("Introduceti numarul de televizoare");
    scanf("%d",&n);
    v=(televizor*)malloc(sizeof(televizor)*(n));
    for(int i=0;i<n;i++)
    {
        printf("Introduceti marca televizorului:");
        char aux[20];
        scanf("%s",&aux);
        (v[i]).marca=(char*)malloc(sizeof(char)*(strlen(aux)+1));
        strcpy(v[i].marca,aux);
        printf("Introduceti Diagonala televizorului:");
        scanf("%d",&v[i].diagonala);
    }
}
```

# S02 - Citire matrice



# S02 – Citire matrice

```
void citireMatrice(televizor** &matrice, int &linii, int &coloane)
{
    printf("numarul de linii:");
    scanf("%d",&linii);
    printf("numarul de coloane:");
    scanf("%d",&coloane);

    matrice=(televizor**)malloc(sizeof(televizor*)*linii);
    for(int i=0;i<linii;i++)
    {
        matrice[i]=(televizor*)malloc(sizeof(televizor)*coloane);
    }

    for(int i=0;i<linii;i++)
    {
        for(int j=0;j<coloane;j++)
        {
            char aux[20];
            printf("Introduceti marca televizorului:");
            scanf("%s",&aux);
            matrice[i][j].marca=(char*)malloc((strlen(aux)+1)*sizeof(char));
            strcpy(matrice[i][j].marca,aux);
            printf("Introduceti diagonala televizorului:");
            scanf("%d", &matrice[i][j].diagonala);
        }
    }
}
```

# S02 - Matrice in Zig Zag

	4	5			
	3	2	6	3	
	2	6	4		
	6	1			

# S02 – Matrice in Zig Zag

```
void citireMatriceZigZag(televizor** &matrice, int &linii, int* &coloane)
{
    printf("numarul de linii:");
    scanf("%d",&linii);
    matrice=(televizor**)malloc(sizeof(televizor*)*linii);
    coloane=(int*)malloc(sizeof(int)*linii);

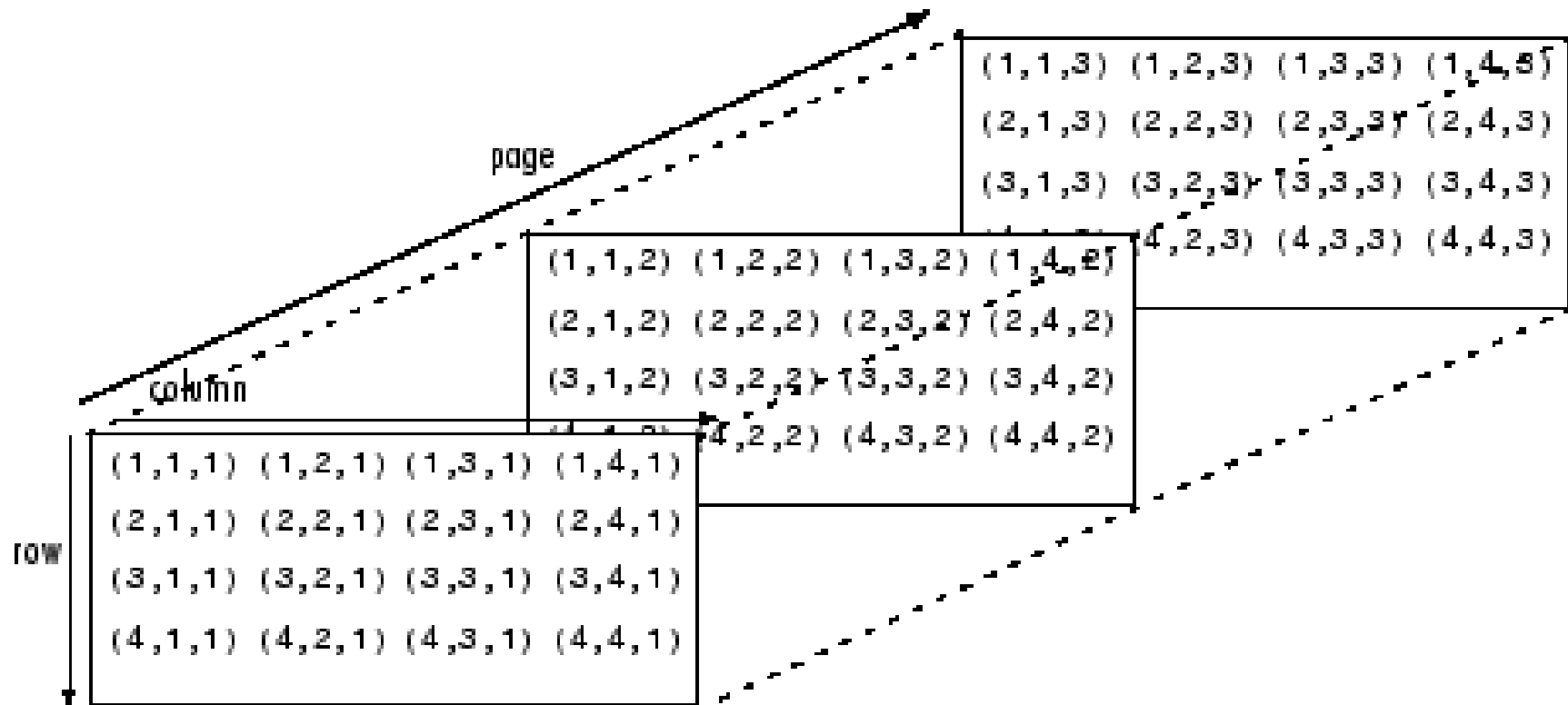
    for(int i=0;i<linii;i++)
    {
        printf("numarul de coloane pentru linia %d:",(i+1));
        scanf("%d",&coloane[i]);
        matrice[i]=(televizor*)malloc(sizeof(televizor)*coloane[i]);

        for(int j=0;j<coloane[i];j++)
        {
            char aux[20];
            printf("Introduceti marca televizorului:");
            scanf("%s",&aux);
            matrice[i][j].marca=(char*)malloc((strlen(aux)+1)*sizeof(char));
            strcpy(matrice[i][j].marca,aux);
            printf("Introduceti diagonala televizorului:");
            scanf("%d", &matrice[i][j].diagonala);
        }
    }
}
```

# S02 - Matrice in Zig Zag

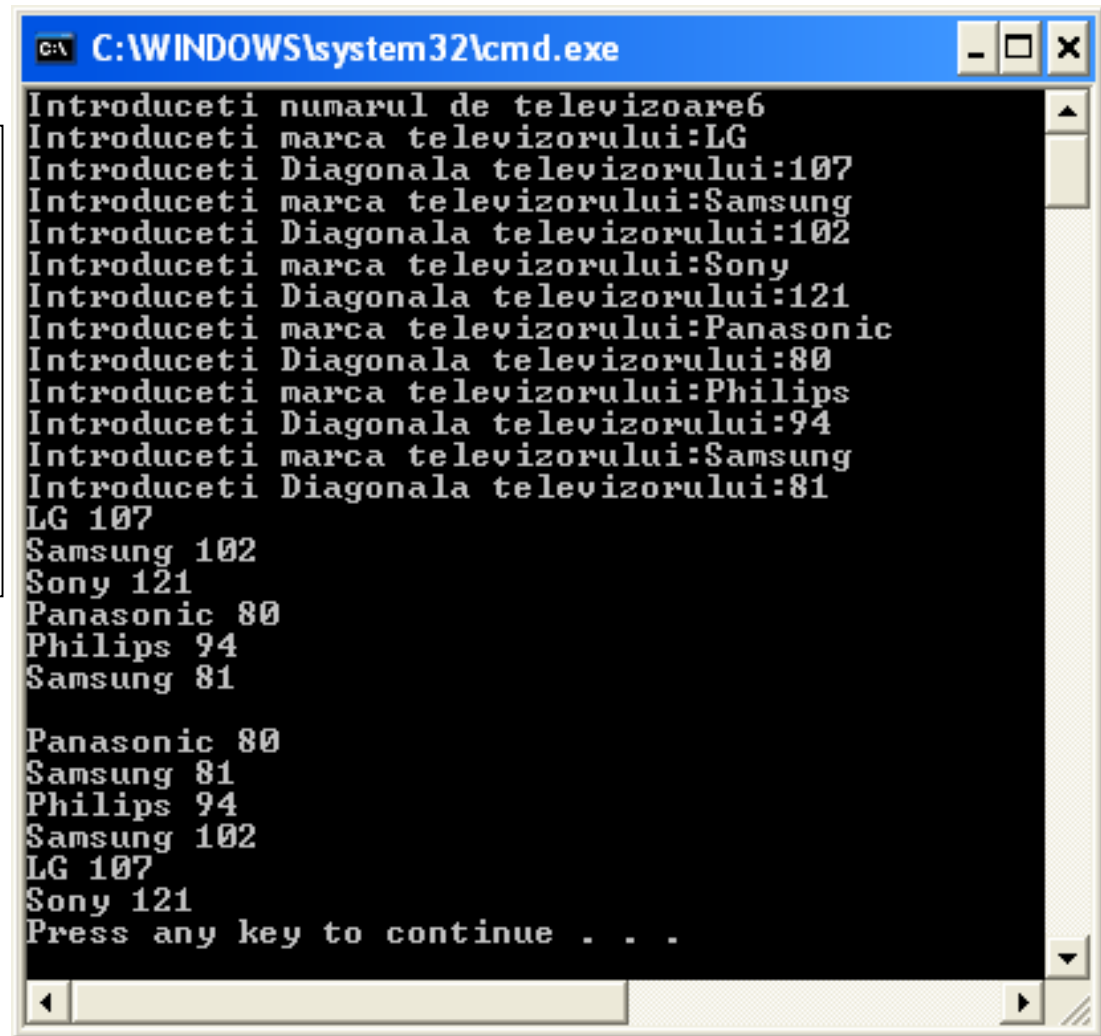
```
void afisareMatriceZigZag(televizor **matrice, int n, int *m)
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<m[i];j++)
        {
            printf("%s %d \t\t", matrice[i][j].marca, matrice[i][j].diagonala);
        }
        printf("\n");
    }
}
```

# S02 - Tablou n-dimensional



# S02 - Sortare vector

```
televizor* vector;  
int lungime;  
citireVector2(vector, lungime);  
afisareVector(vector, lungime);  
printf("\n");  
sortareVector(vector, lungime);  
afisareVector(vector, lungime);
```



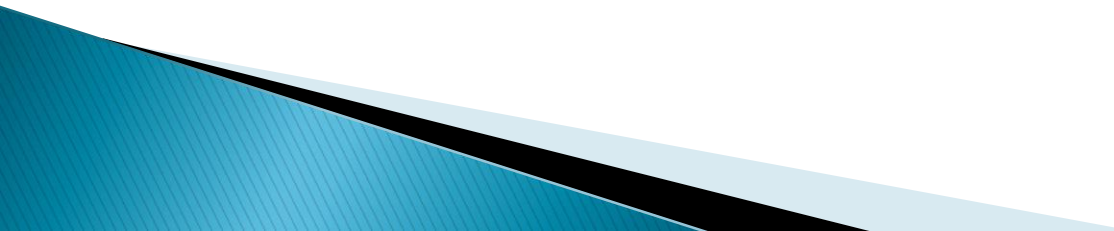
```
C:\WINDOWS\system32\cmd.exe  
Introduceti numarul de televizoare6  
Introduceti marca televizorului:LG  
Introduceti Diagonala televizorului:107  
Introduceti marca televizorului:Samsung  
Introduceti Diagonala televizorului:102  
Introduceti marca televizorului:Sony  
Introduceti Diagonala televizorului:121  
Introduceti marca televizorului:Panasonic  
Introduceti Diagonala televizorului:80  
Introduceti marca televizorului:Philips  
Introduceti Diagonala televizorului:94  
Introduceti marca televizorului:Samsung  
Introduceti Diagonala televizorului:81  
LG 107  
Samsung 102  
Sony 121  
Panasonic 80  
Philips 94  
Samsung 81  
  
Panasonic 80  
Samsung 81  
Philips 94  
Samsung 102  
LG 107  
Sony 121  
Press any key to continue . . .
```



# S02 – Sortare vector

```
void sortareVector(televizor* &v,int n)
{
    for(int i=0;i<n-1;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(v[i].diagonala>v[j].diagonala)
            {
                char aux[20];
                strcpy(aux,v[i].marca);
                free(v[i].marca);
                v[i].marca=(char*)malloc(sizeof(char)*(strlen(v[j].marca)+1));
                strcpy(v[i].marca,v[j].marca);
                free(v[j].marca);
                v[j].marca=(char*)malloc(sizeof(char)*(strlen(aux)+1));
                strcpy(v[j].marca,aux);
                int temp=v[i].diagonala;
                v[i].diagonala=v[j].diagonala;
                v[j].diagonala=temp;
            }
        }
    }
}
```

# Tema

- ▶ Să se realizeze în limbajul C jocul de avioane, în care utilizatorul se joacă cu calculatorul.
  - ▶ Să se folosească o matrice pentru câmpul de luptă.
  - ▶ Utilizatorul alege la început cât de mare să fie câmpul de luptă. Atât numărul de linii cât și numărul de coloane.
  - ▶ Apoi programul construiește în memorie câmpul de luptă pe care pune un avion.
- 

# Tema

- ▶ Apoi i se cere utilizatorului să introducă coordonatele unde va ataca pe câmpul de lupta. Dacă lovește avionul marchează cu valoarea 2, dacă lovește în aer, se marchează cu 1. Dacă lovește avionul în cap, este desenat tot avionul pe câmpul de luptă cu valoarea 3 – avion doborât.
- ▶ Designul avionului e la alegerea fiecăruia

```
C:\WINDOWS\system32\cmd.exe
4
6
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

# Hint tema

- ▶ Pentru reținerea coordonatelor avioanelor pe câmpul de luptă se poate folosi o structura.
- ▶ Astfel vom avea o matrice de articole.
- ▶ Structura va avea 3 câmpuri:
  - Valoarea desenată pentru utilizator;
  - Valoarea din memorie, dacă acolo este o parte de avion sau vârful (capul) avionului, sau dacă este aer;
  - Valoarea care să prezinte faptul că acolo a lovit deja în acel sector.
- ▶ Să se contorizeze numărul de inserări ale utilizatorului până doboară avionul de pe câmpul de luptă. La final i se afișează utilizatorului din câte încercări a doborât avionul.