

## Practical Work No. 6

### Two-dimensional arrays

#### Goals

- Manipulating two-dimensional arrays
- Declaring two-dimensional arrays
- Initializing two-dimensional arrays

#### I. Declaration

##### Syntax:

`<simple type> <Table name> [<dimlig>] [<dimcol>];`

##### Examples:

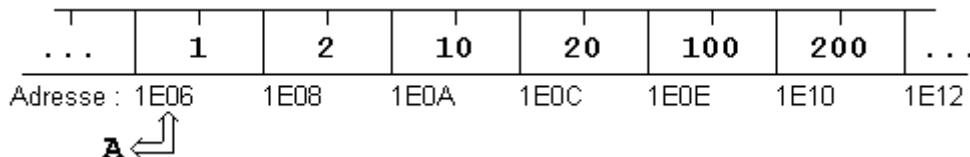
```
int A [10] [10] ;
float B [2] [10] ;
char C [10] [20] ;
```

#### II. Memorization

As with one-dimensional arrays, the name of an array is the address of the first element in the array (i.e., the address of the first row of the array). The components of a two-dimensional array are stored row by row in memory.

##### Examples:

```
int A [3][2] = {{1,2},{10,20},{100,200}};
```



```
int B[3][4] = {{1,2,3,4},{10,20,30,40},{100,200,300,400}};
```

Reservation of  $3*4*2 = 24$  bytes

##### Noticed :

If the number of lines L is not explicitly specified during initialization, the computer automatically reserves the necessary number of bytes.

```
int B[][10] = {{ 0,10,20,30,40,50,60,70,80,90},{10,11,12,13,14,15,16,17,18,19},
```

```
{ 1,12,23,34,45,56,67,78,89,90 }};
```

Reservation of  $3*10*2 = 60$  bytes

#### III. Access to the components of a matrix

Consider an array A of dimensions R (rows) and C (Columns).

- ❖ The table indices vary from 0 to R-1, respectively from 0 to C-1.
- ❖ the component of the N<sup>th</sup> row and M<sup>th</sup> column is noted:

**A[N-1][M-1]**

#### IV. Loading a table

```
void main (){
    int A [5] [10];
    int i, j;
    /* For each line...*/
    for (i=0; i<5; i++) {
        for (j=0; j<10; j++) {
            printf ("A[%d][%d]:", i ,j);
            scanf ("%d", &A[i][j]);
        }
    }
}
```

#### V. Displaying the contents of a table

```
void main(){
    int A[5][10];
    int i, j;
    /* For each line...*/
    for (i=0; i<5; i++) {
        for (j=0; j<10; j++)
            printf("%7d", A[i][j]);
        /* Line break */
        printf("\n");
    }
}
```

#### VI. Work requested

##### Exercise 1

Write a program that reads the dimensions R and C of a two-dimensional array T of type int (maximum dimensions: 50 rows and 50 columns). Fill the array with values entered from the keyboard and display the array and the sum of all its elements.

##### Exercise 2

Write a program that reads the dimensions R and C of a two-dimensional array T of type int (maximum dimensions: 50 rows and 50 columns). Fill the array with values entered from the keyboard and display the array and the sum of each row and column using only a helper variable for the sum.

##### Exercise 3

Write a program that transfers a two-dimensional array M (maximum dimensions: 10 rows and 10 columns) into a one-dimensional array L as follow.

##### Example:

a b c d  
e f g h ==> abcdefghijkl  
i j k l