BADJI MOKHTAR UNIVERSITY
ANNABA

C Arrays

BY

Dr. Samira LAGRINI

Academic year:2023/2024

Introduction

- Simple variables correspond to memory cells which store one value at a time.
- For several values, the use of distinct variables would be difficult to manage and manipulate.

- Arrays are data structures used to store multiple values in a single variable, instead of declaring separate variables for each value.

What is Array in C?

❑ An array in C is a fixed-size collection of similar data items stored in contiguous memory locations

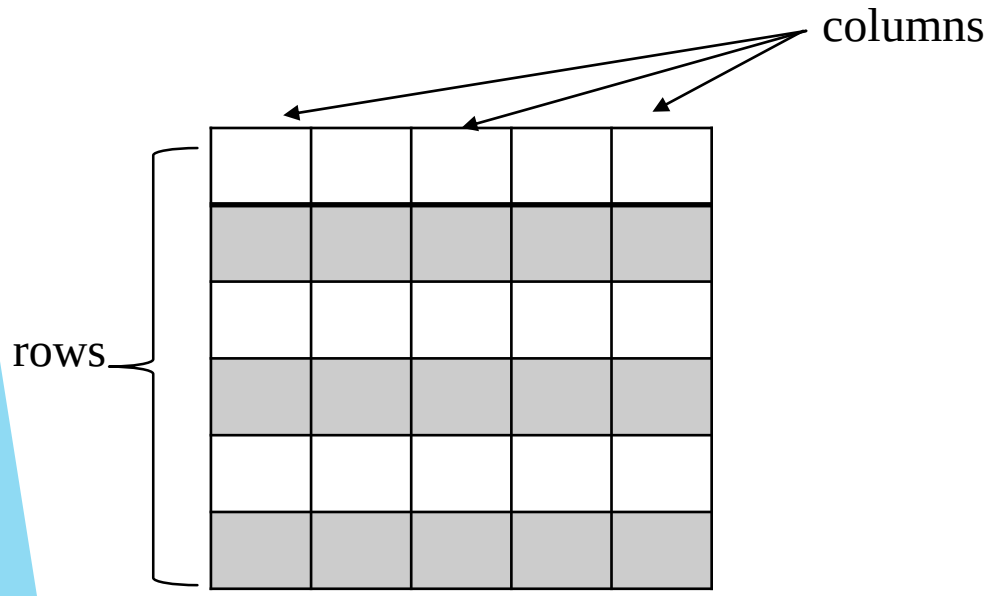
5	2	…. …. …..	3	12

Array elements can be:

- **Simple data type**: int, char, float, ...
- pointers, structures, etc

What is Array in C?

- An array composed of simple type data is called single-dimensional array (or vector)
- An array composed of vectors is called a matrix



matrix

Single dimensional array

single-dimensional arrays

Declaration of Array in C

- In C, the declaration of a one-dimensional array is done as follows:

Type array's name [size] ;

the type of array elements (int, float, char,…)

A positive integer constant that specifies the number of elements in the array

Declaration of Array in C (2)

Examples:

- **float** tab [20];
- **int** table [10];
- **char** chaine [10];

⟹ For clarity, it is recommended to declare a constant that specifies the Size of the array.

Example

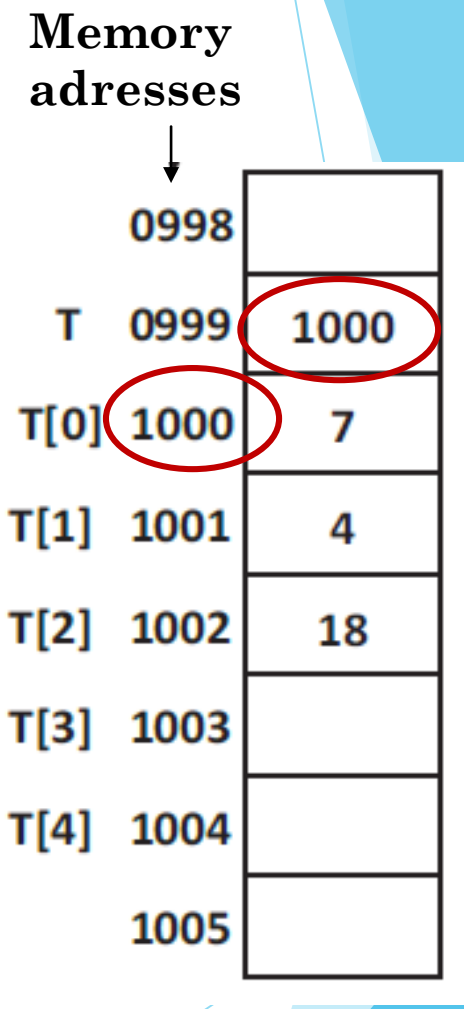
```
#define taille 10
main()
{
   int tab[taille];
    ...
}
```

Storing an array in the memory

- In C, the name of an array contains the address of the first element of the array.

- The addresses of the other components are calculated automatically, relative to this address.

- If an array has N components and if the declared type of the components requires M bytes, the memory reserved for this array is N×M bytes.



Array initialization in C

It is possible to initialize an array when declaring it, by indicating the list of values

Example

```
int tab[4] = {1, 1, 31, 4};
```

- If the number of data in the initialization list is less than the dimension of the array, only the first elements will be initialized.

  The other elements will be set to zero.

Array initialization in C (2)

❑ It is also possible to omit the size of the array since the compiler is able to determine it by counting the number of elements present in the initialization list

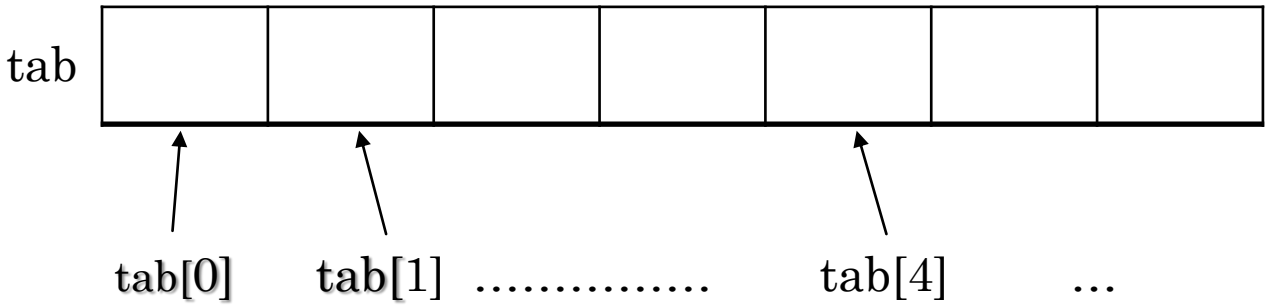
Example:

```
float vec[ ]={1,5,6}
```

The three elements of the array are initialized to 1, 5 and 6, the size of the array will automatically be set to 3.

Accessing Array Elements

- An array element is accessible by its position (indice) in the array
- Indices of an array range from 0 to size - 1.



- Indices are used to access any element of the array in the following way:

array_name[index]

Example:

Tab [3]=22;

Input Array Elements

- Array values can be stored by taking input from the user and storing them in the array

Example:

```
int tab[N], i;
for (i = 0; i < N; i++)
   scanf ("%d", &tab[i]);
```

Output Array Elements

The display of a table is always done element by element using loops.

Example:

```
    int tab[N], i;
      ……..
   for (i = 0; i < N; i++)
      printf ("%d\t",  tab[i]);
}
```

EXERCISE 1

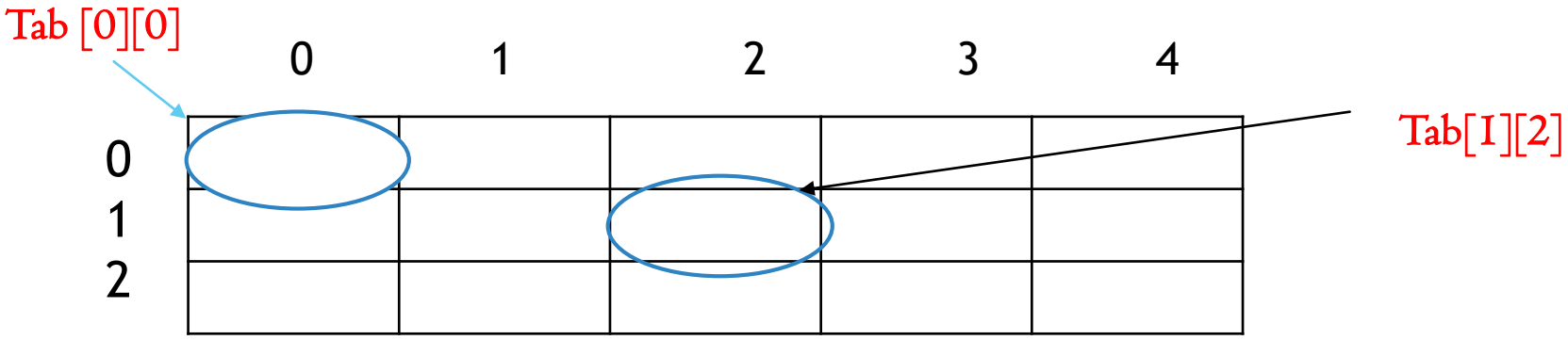
Write a C program that :
- Reads an integer array of 10 elements
- Calculates the sum of its elements
- displays the table

```
#include <stdio.h>
#define N 10
main()
{
   int tab[N], i, som;
/*Remplir le tableau*/
for (i = 0; i < N; i++)
      scanf("%d", &tab[i]);
/* somme des éléments du tableau*/
Som=0;
   for (i = 0; i < N; i++)
      som=som+tab[i];

Printf ("la somme des éléments =%d", som);
}
```

Two-dimensional arrays

- A two-dimensional array also called a matrix can be represented by a set of boxes organized into rows and columns and schematized as follows:



- A matrix element is accessible by two indices. The first corresponds to the row number and the second to the column number

Declaration

- *The declaration of a 2-dimensional array (matrix) is done as follows:*

type matrix_name[row][column] ;

Example :

int A[10][10]; tableau de 10×10 entiers de type int

float B[5][4]; tableau de 5×4 décimaux de type float

char C[2][25]; tableau de 2×25 caractères

Initialisation

- It is possible to initialize a matrix when declaring it, by indicating the list of values,

Example :

int A[3][2]= {{1,2},{10,20},{100,200}}; 3x2 integer matrix,

- If the number of rows or columns is not explicitly declared, the computer automatically reserves the necessary number of bytes.

Displaying a matrix

- To display a matrix we use two nested loops and two indices (for example i and j) to traverse the rows and columns

Example:

```
int A[5][10]
for(i=0; i<5; i++) /* boucle sur les lignes*/
{
    for(j=0; j<10; j++) /* boucle sur les  colonnes */
     printf("%d\t", A[i][j]);
}
printf ("\n");
}
```

Input matrix Elements

- As for display, to fill a matrix with values entered on the keyboard, you must use two nested loops and two indices to traverse the rows and columns

Example:

```
int A[5][10]
for(i=0; i<5; i++) /* boucle sur les lignes*/
    for(j=0; j<10; j++) /* boucle sur les  colonnes */
     scanf("%d", &A[i][j]);
```

EXERCISE

Write a C program that asks the user to fill in two 3*3 integer matrices and then displays their product.

Solution

```
#include <stdio.h>
main()
{
  int mat1[3][3], mat2[3][3], produit[3][3] ;
  int i, j, k, x, s ;
  printf("          input the elements of the 1st matrix          :\n") ;
  for(i=0;i<=2;i++)
     for(j=0;j<=2;j++) scanf("%d", &mat1[i][j]);
  printf("          input the elements of the 2 nd matrix         :\n") ;
   for(i=0;i<=2;i++)
      for(j=0;j<=2;j++) scanf("%d", &mat2[i][j]);
   for(i=0;i<=2;i++) for(j=0;j<=2;j++) {
       s = 0 ;
       for(k=0;k<=2;k++) s = s + mat1[i][k] * mat2[k][j] ;
       produit[i][j] = s ;
     }
   printf("              Product result                \n") ;
   for(i=0;i<=2;i++)
      for(j=0;j<=2;j++) printf("%d\n", produit[i][j]) ;
}
```