

Higher School of Applied Sciences of Tlemcen
1-year computer science course



Arrays in C

Part 1

Presented by:
Dr. Imane NEDJAR

Part 1

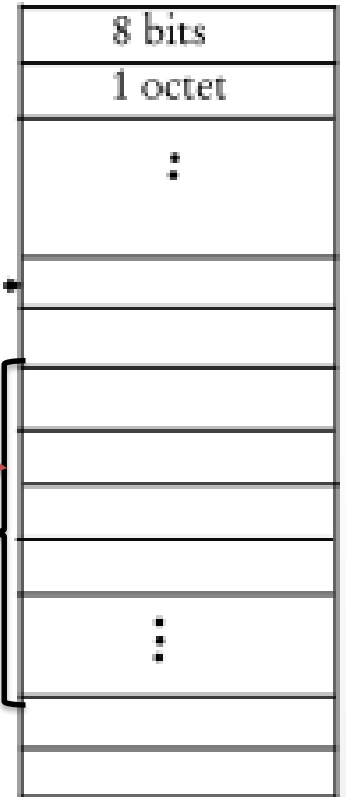
Overview of Arrays

Array

An array is a **variable** consisting of data **of the same type**, stored in contiguous memory locations, one after another.

In essence, an array is a sequence of memory cells, each of uniform size.

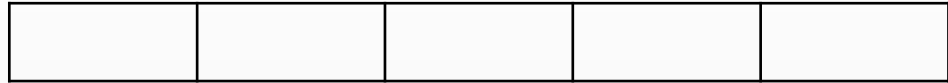
The elements contained within the array can be of basic data types, including int, char, float, long, double, and various others.



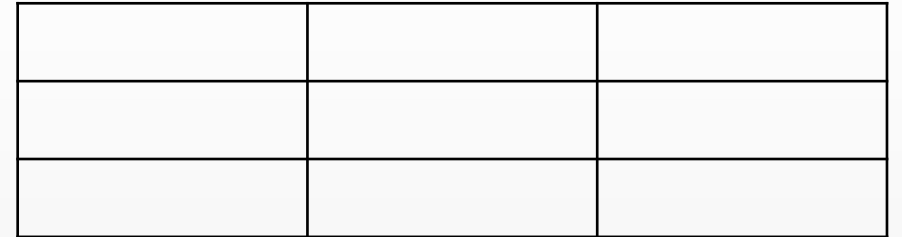
Array

Array

One-dimensional array

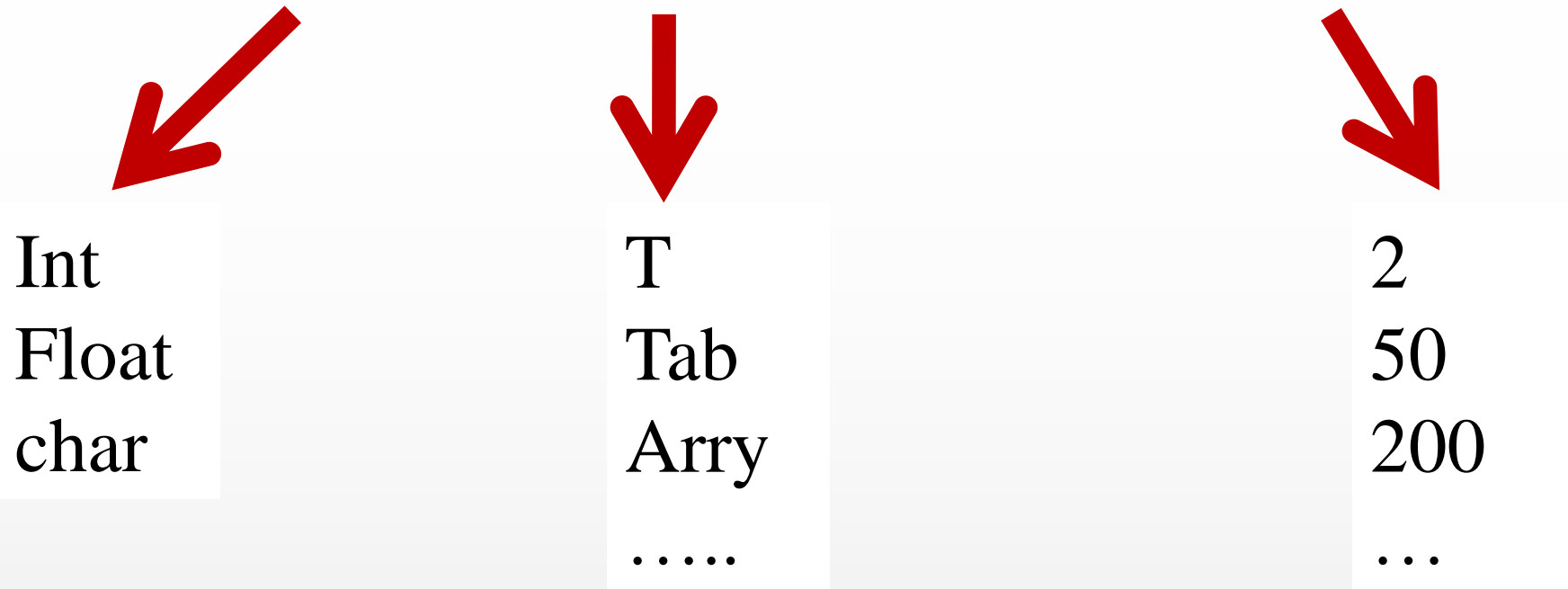


Two-dimensional array



Declaration

Type Name_of_the_array [**Number of elements**]



Example: **int** tab[100];

char name [10];

float moy[300]

- **Type** : specifies the data type of the elements that the array will contain.
- **Name of the array** : is the chosen name for the array, adhering to the same naming conventions as a variable.
- **Number of elements** : is an integer that determines the number of elements the array should have.

Manipulate the elements

Any element within the array (identified by its array name and index) can be manipulated just like a variable. This means you can perform operations on elements of the array.

Tab	10	20	30	40	50
	0	1	2	3	4

Let's define an array of 5 integers:

```
int Tab[5]
```

```
Tab[1] = 20;
```

```
Tab[4] = Tab [1] + Tab[2];
```

Reading an array

```
for (i=0; i<N; i++)  
    { scanf ("%d", &T[i]); }
```

A red oval with a thin blue border containing the white number 1.

```
for (i=0; i<N; i++)  
    { printf("T[%d]=",i);  
      scanf ("%d", &T[i]);  
    }
```

A red oval with a thin blue border containing the white number 2.

Display an array

```
for (i=0; i<N; i++)  
{ printf("%d ", T[i]); }
```



```
for (i=0; i<N; i++)  
{ printf("T[%d]= %d", i, T[i]); }
```



Reading and displaying an array

```
#include <stdio.h>
```

```
int N, i, T[20];
```

```
int main ()
```

```
{
```

```
    printf("Give the number of elements N:");
```

```
    scanf("%d",&N) ;
```

```
    For (i=0; i<N; i++)
```

```
    { printf("T[%d]=", i);
```

```
      scanf ("%d", &T[i]);
```

```
    }
```

```
    For (i=0; i<N; i++)
```

```
    printf ("T[%d]= %d \n", i, T[i]);
```

```
return 0 ; }
```

Initialize the elements

```
int Tab[5] = { 10, 20,30,40,50};
```

The following instruction is used to set all elements of the array to zero:

```
int Tab[5] = {0};
```

Initialize the elements

```
#include <stdio.h>

int N, i;

int T[8]={3, 5, 8, 21, 77, 2, 4, 34};

int main ()
{
    printf(" The elements of the array :");
    for (i=0; i<8; i++)
        printf("T[%d]= %d\n", i, T[i]);

    return 0 ;}
```

Exercise 1

Create a program that accomplishes the following tasks:

- Read the values of elements an array T consisting of N real numbers.
- Compute the sum of the elements in T.
- Calculate the average of the elements in T.
- Display the values of all elements in T, as well as the sum and the average.

Exercice 2

Create a program that performs the following tasks:

- Read values of elements in an array T consisting of N real numbers.
- Search for the position of a user-entered element.
- Calculate the number of occurrences of a user-entered element.

Multidimensional arrays are arrays with rows and columns.

For example, a two-dimensional array (3 rows, 3 columns).

Is stored in memory as follows



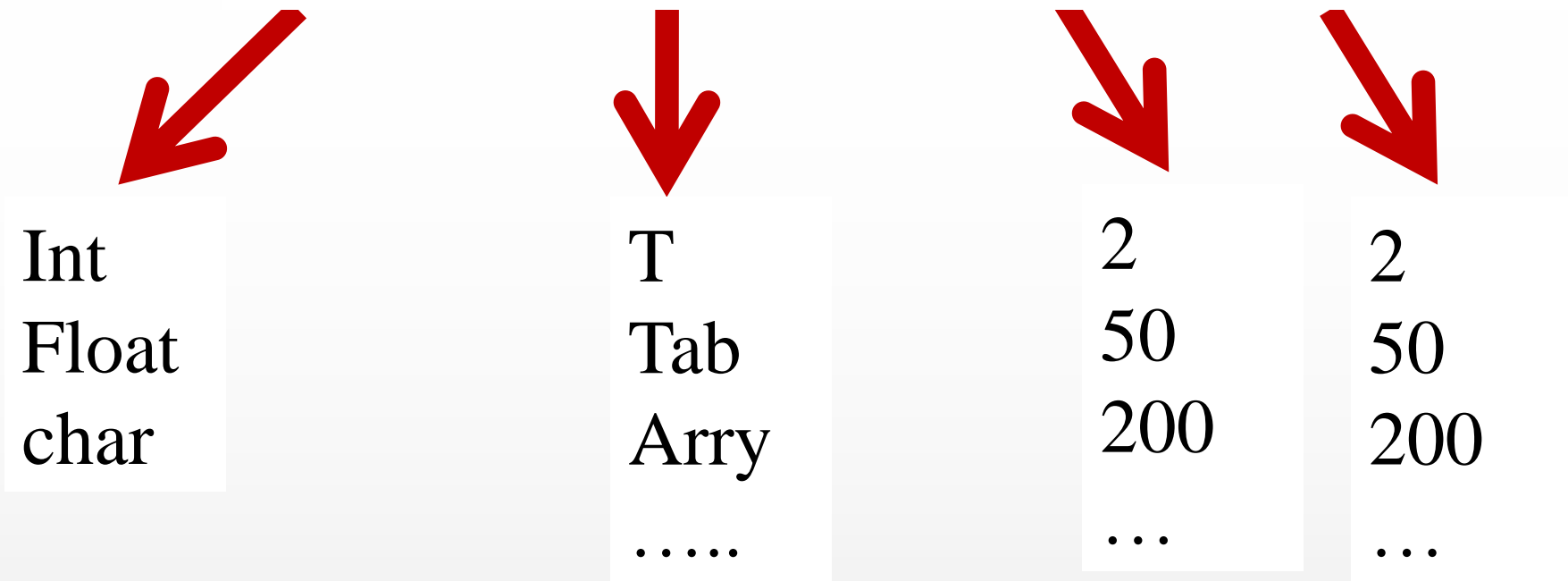
Multidimensional arrays are arrays with rows and columns.

For example, a two-dimensional array (3 rows, 4 columns).

Tab[0][0]	Tab[0][1]	Tab[0][2]	Tab[0][3]
Tab[1][0]	Tab[1][1]	Tab[1][2]	Tab[1][3]
Tab[2][0]	Tab[2][1]	Tab[2][2]	Tab[2][3]

Declaration

Type Name_of_the_array [Nb-ele] [Nb-ele]



Example: **int** tab[100][20]; **char** name[10] [20]; **float** moy[30] [14]

Initialize the elements

- **Individual initialization of each element**

```
tab[0][0] = 2; tab[0][1] = 3
```

- **Initialization through loops**

```
int i,j;  
  
for (i=0; i<=2; i++){  
  
for (j=0; j<=3; j++){  
  
tab[i][j] = 0; } }  
}
```

Reading a matrix

```
for (i=0; i<nb_row; i++)  
{for (j=0; j<nb_col; j++)  
    {scanf ("%d", &T[i][j]); } }
```

1

```
for (i=0; i<nb_row; i++)  
{for (j=0; j<nb_col; j++)  
    { printf("T[%d][%d]=",i,j);  
      scanf ("%d", &T[i][j]); } }
```

2

Displaying an array

```
for (i=0; i<nb_row; i++)  
{for (j=0; j<nb_col; j++)  
    { printf("%d",M[i][j]); } }
```



```
for (i=0; i<nb_row; i++)  
{for (j=0; j<nb_col; j++)  
    { printf("M[%d][%d]=%d",i,j,M[i][j]); } }
```



Reading and displaying an array

```
#include <stdio.h>
```

```
int N, P, i, j, M[20][10];
```

```
int main ()
```

```
{ printf("Provide the number of rows N :");
```

```
scanf("%d",&N) ;
```

```
printf("Provide the number of columns P :");
```

```
scanf("%d",&P) ;
```

```
for (i=0; i<N; i++)
```

```
    for (j=0; j<P; j++)
```

```
    { printf("M[%d][%d]=", i,j);
```

```
      scanf ("%d", &M[i][j]); }
```

```
for (i=0; i<N; i++)
```

```
    for (j=0; j<P; j++)
```

```
        printf("M[%d][%d]=%d \n", i,j,M[i][j]) ;
```

```
return 0 ;}
```

Exercise

Create a program that accomplishes the following tasks:

- Read the real values of elements in a matrix M with N rows and P columns.
- Calculate the sum of the elements in M .
- Display all the values of elements in M and the sum.