TP7: The Arrays

1. The vectors: (one-dimensional arrays):

1.1. Declaration :

We declare vectors in the C++ language as follows:

Type_Elements ***name***_vector [Vector_Size];

Examples:

int v1[100];
float vect [50];
char tab3[20]; // A string

v1 is a vector of 100 integers, vect is a vector of 50 reals, and tab3 is a vector of 20 characters.

v1:

6	- 5	90	- 45	.	.	.	.	.	13
i=0	i=1	i=2	i=3		.	.	.	.	i=99

vect:

6	- 5.20	90	- 45.05	.	.	.	13
i=0	i=1	i=2	i=3		.	.	i=49

tab3:

U	NOT	&	6	(	M	$	.	.	W
i=0	i=1	i=2	i=3		.	.	.	.	i=19

- ➢ Each box in a vector has a number. The number of the first box is **0** and the number of the last box is **n-1** (n is the size of the vector).
- ➢ The box number is also called index or position.

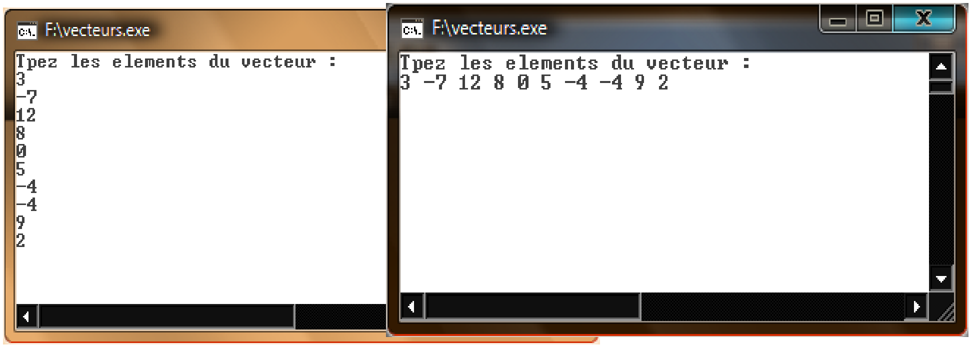
Example : V1 [2] = 90 Vect [3] = - 45.05 tab3 [2] = &

1.2. Reading a vector:

Consider a vector with integer elements of size 10: int v[10]
To fill this vector we use the **cin** instruction as follows:

```
cin>> v[0];
cin>> v[1];
cin>> v[2];
.                      more efficient                 for (i=0;i<=9;i=i+1)
.          Or     ------------------------->            cin>>v[i] ;
 .                     using a loop
cin >>v[9] ;
```

The vector data can be entered vertically or horizontally as follows:



```
Tpez les elements du vecteur :
3
-7
12
8
0
5
-4
-4
9
2
```

```
Tpez les elements du vecteur :
3 -7 12 8 0 5 -4 -4 9 2
```

By separating the values by spaces, or by line break (Enter button).

1.3. Writing a vector:

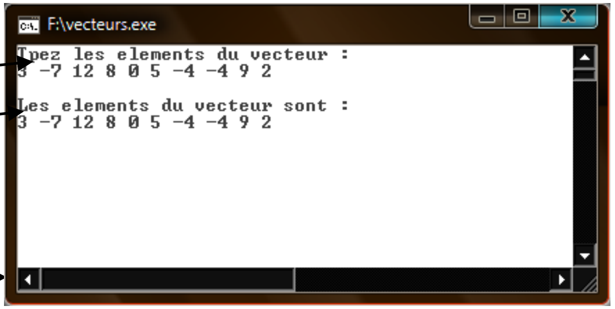
To display the preceding vector v we use the **cout** instruction as follows:

```
cout<<v[0];
cout <<v[1];
cout <<v[2];
 .                     using            for (i=0; i<=9; i=i+1)
.          Or     ------------->                cout <<v[i];
.                      the loopfor
cout <<v[9];
```

We can write the vector data vertically, or horizontally as follows:

Reading

Writing



```
Tpez les elements du vecteur :
3 -7 12 8 0 5 -4 -4 9 2

Les elements du vecteur sont :
3 -7 12 8 0 5 -4 -4 9 2
```

for(i=0;i<=9;i=i+1)

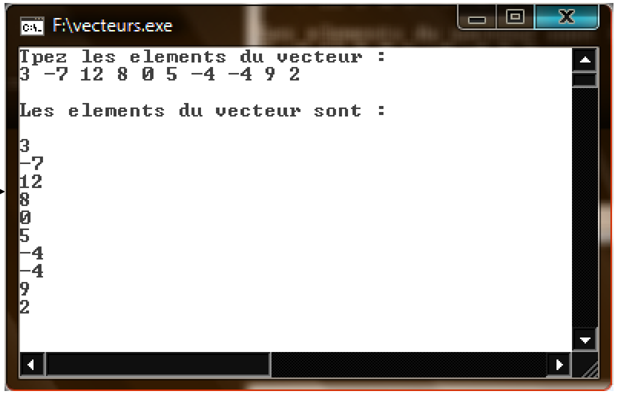
cout<<v[i]<<" "

Space

for(i=0;i<=9;i=i+1)

cout<< "\n"<< v[i];

\n: line break



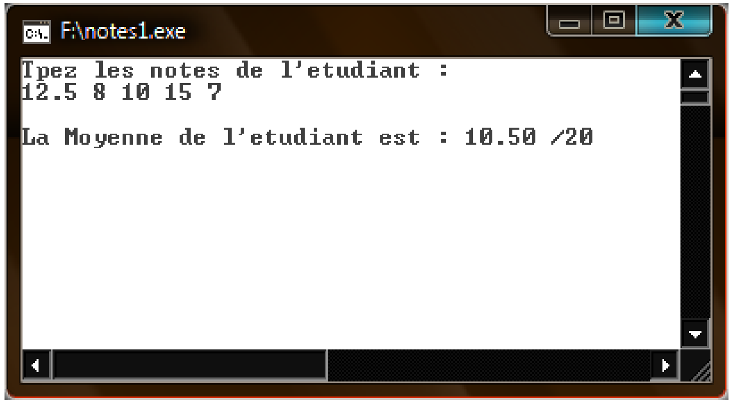
```
Tpez les elements du vecteur :
3 -7 12 8 0 5 -4 -4 9 2

Les elements du vecteur sont :

3
-7
12
8
0
5
-4
-4
9
2
```

Exercise 1:

Write a C++ program that allows you to enter a student's grades in 5 modules and displays the student's average.



```
Tpez les notes de l'etudiant :
12.5 8 10 15 7

La Moyenne de l'etudiant est : 10.50 /20
```

2. The matrices: (two-dimensional arrays)

2.1 Declaration of a matrix:

We declare matrices in the C++ language as follows:

Type_Elements ***name***_matrix [dimension1][dimension2];

Max number of lines   Max number of columns

Examples1:

int m1[100][100];

float mat [50][10];

char tab[20][30];

- **m1**is an integer matrix of 100 rows and 100 columns,
- **mat** is a real matrix of 50 rows and 10 columns,
- **tab** is a character matrix of 20 rows and 30 columns.

Examples2:

An integer matrix of 5 rows and 4 columns:

int mat[5][4];

Column indices

	0	1	2	3
0	*4*	*12*	*33*	*7*
1	*0*	*7*	*12*	*11*
2	*6*	*5*	*4*	*13*
3	*88*	*63*	*45*	*0*
4	*6*	*99*	*39*	*3*

Line indices

- Rows and columns start from index 0.
- In the previous matrix: **mate[2][1]** = **5**

2.2 Reading the matrices:

Let M be an integer matrix of 5 rows and 3 columns: int M [5][3]

To read this matrix we must use two nested loops with the **cin** instruction as follows:

```
for (i=0; i<=4; i++)
 for (i=0; j<=2; j++)
     cin >>M[ i ][ j ] ;
```

Noticed:

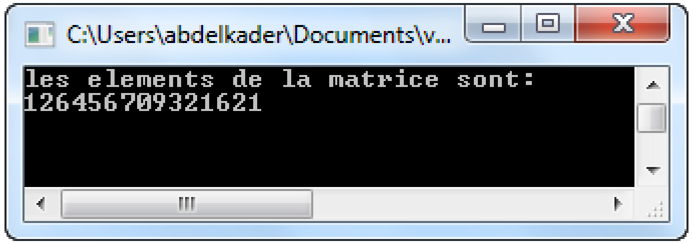
- The first loop for the row index, the second for the column index.
- The index value starts from **0** to **dimension -1** (in our example **5-1** for rows and **3-1** for columns).

2.2 Writing matrices:

To display the values of the previous matrix M we use two loops like reading but with the **cout** instruction as follows:

```
for(i=0; i<=4; i++)
 for(i=0; j<=2; j++)
   cout <<M[ i ][ j ];
```

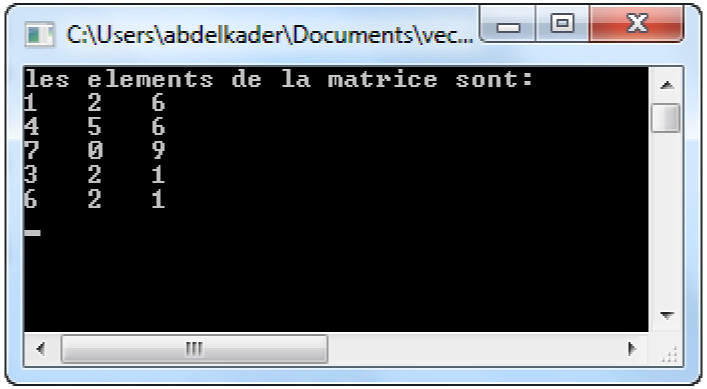
The display of values will be as follows:



```
les elements de la matrice sont:
126456709321621
```

To adapt the display of values and make it more readable, we must modify the code as follows:

```
for(i=0; i<=4; i++){
 for(i=0; j<=2; j++)
    cost << M[ i ][ j ] << " ";
 cost<<"\n"
}
```



```
les elements de la matrice sont:
1   2   6
4   5   6
7   0   9
3   2   1
6   2   1
```

Exercise 2:

Write a C++ program that reads two matrices A and B, then calculate and display matrix C; the sum of these two matrices?

Exercise 3:

Create a C++ program that removes duplicates from an array of integers:

Example :

4	1	3	7	3	1	7	9

→

4	1	3	7	9

Exercise 4:

Write a C++ program that reads a matrix and calculates its transpose; the rows of the initial matrix become the columns of the transposed matrix, then displays it.

4	***12***	***33***	***7***
0	***7***	***12***	***11***
6	***5***	***4***	***13***

→

4	0	6
12	7	5
33	12	4
7	11	13