Arrays (Part I+II)

Dr. Chénche

Two-dimensional Arrays (Matrix)

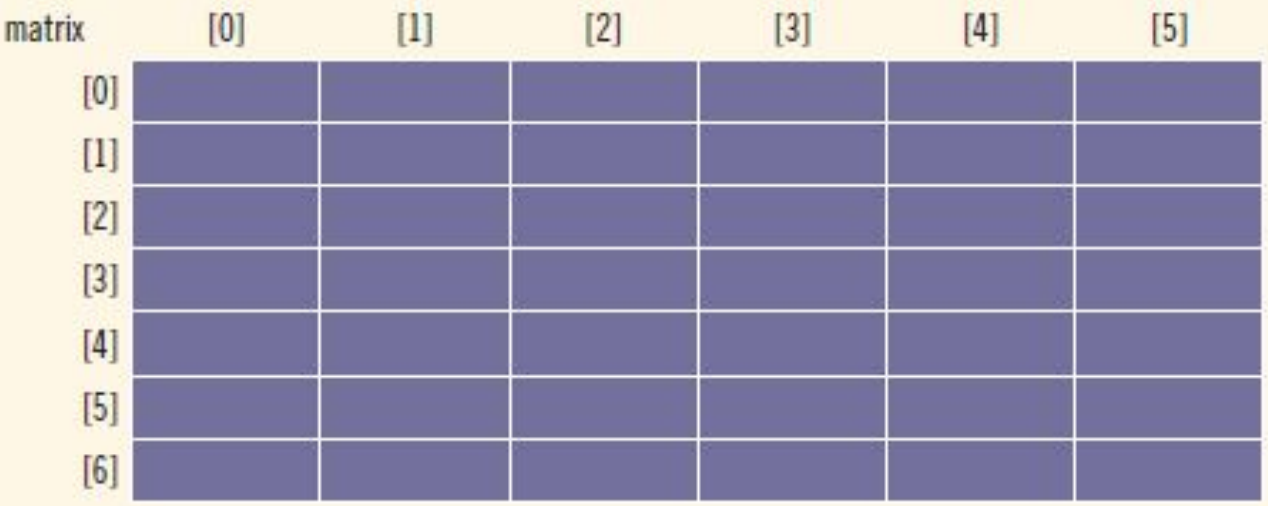
- A matrix is: a collection of a fixed number of components (of the same type) arranged in two dimensions.
- **Declaration syntax:**

```
Data_Type Matrix_Name[Nbline][Nbcol];
```

where **Nbline** and **Nbcol** are **integer values** that specify the number of rows and the number of columns, respectively, in the matrix.

Declaration of Matrix

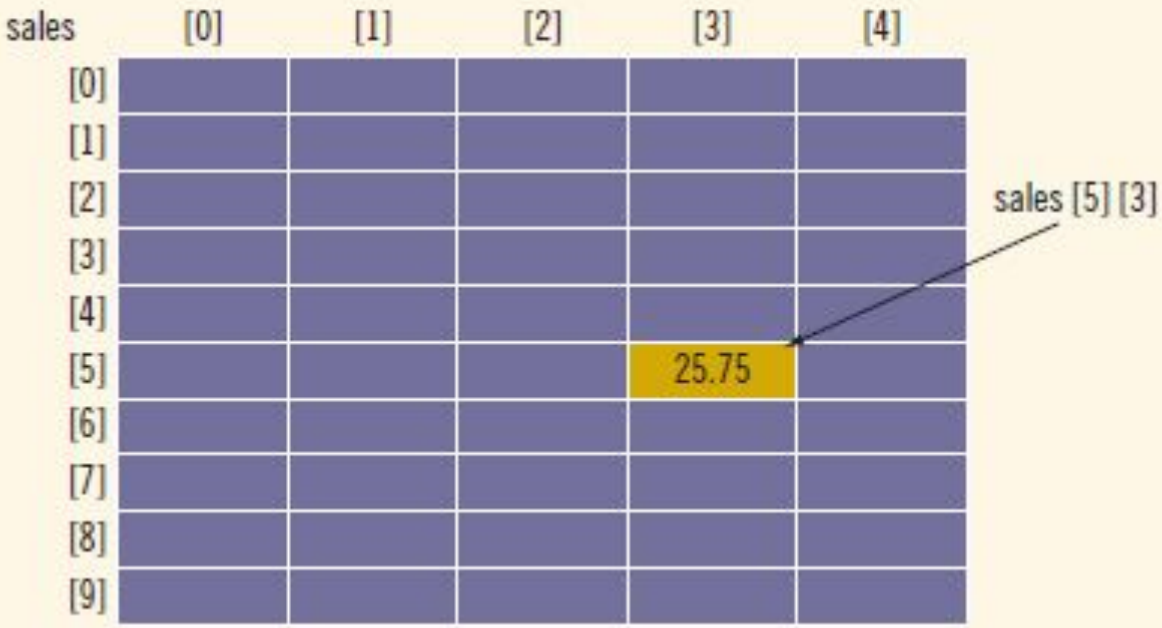
```
int main()
{
int Matrix[7][6];

return 0;
}
```



Accessing Matrix elements

Example :

```
float sales[10][5];
```



Matrix Initialization During Declaration

- A matrix can be initialized when it is declared:

```
int Matrix[4][3]={ {1,2,3},{4,5,6},{7,8,9},{10,11,12} };
```

  - Elements of each row are enclosed within braces and separated by commas.
  - All rows are enclosed within braces.

Processing Matrix

- **Ways to process a Matrix :**
  - Process the entire Matrix
  - Process a particular row of the array, called row processing
  - Process a particular column of the array, called column processing
- Each row and each column of a matrix is a one-dimensional array
  - To process each, use process similar to processing one-dimensional arrays

Initialization

- **To initialize the entire matrix by 0:**

```
for (i=0;i<n;i++)
for (j=0;j<m;j++)
Matrix[i][j]=0;
```

- **To initialize fifth (5) row by 0:**

```
for (j=0;j<m;j++)
Matrix[4][j]=0;
```

Reading (Fill in) a Matrix

- To input data into each component of matrix:

```
int T[100][100],i,j,n=3,m=4;
for (i=0;i<n;i++)
 for (j=0;j<m;j++)
      { printf("T[%d][%d]=",i+1,j+1);
        scanf("%d",&T[i][j]);}
```

Display a Matrix

- To output the components of matrix:

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

Traversals in a matrix

1- Row traversal

- **Row traversal** refers to **visit elements in a matrix row by row**.
- **How does "Row Traversal" Work in a Matrix?**

1- Row traversal

- For a matrix with dimensions **n x m** (n rows and m columns):

- You start at the first row (row 0).

- For each row, you go through each column elements from left to right.

- Once you complete a row, you move to the next row until you finish traversing all rows in the matrix.

1- Row traversal

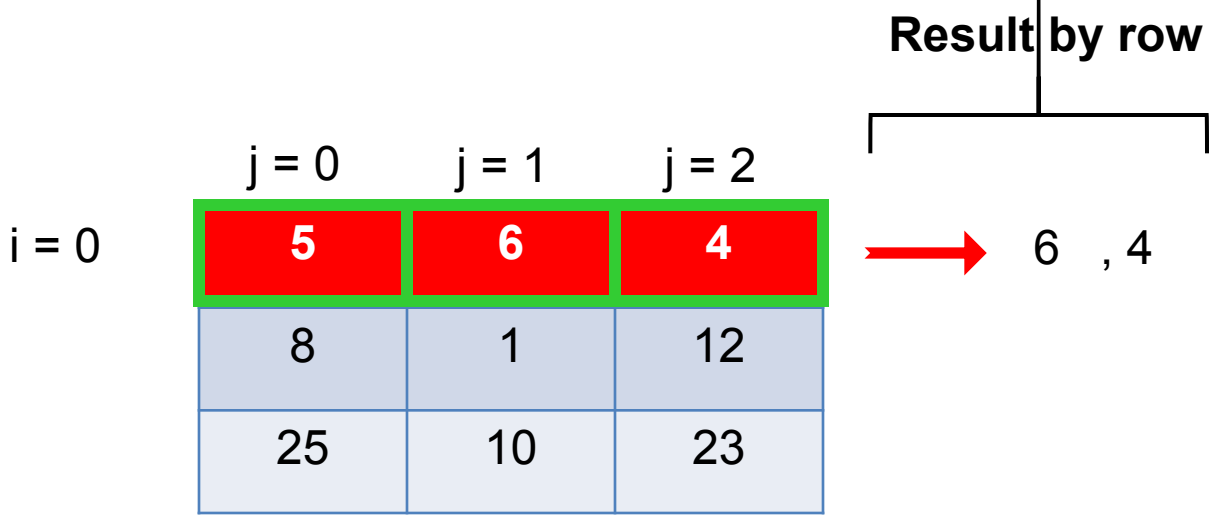
- **Example**

- Traverse each row of a matrix and display **even elements** in each row.

5	**6**	**4**
8	1	12
25	10	23

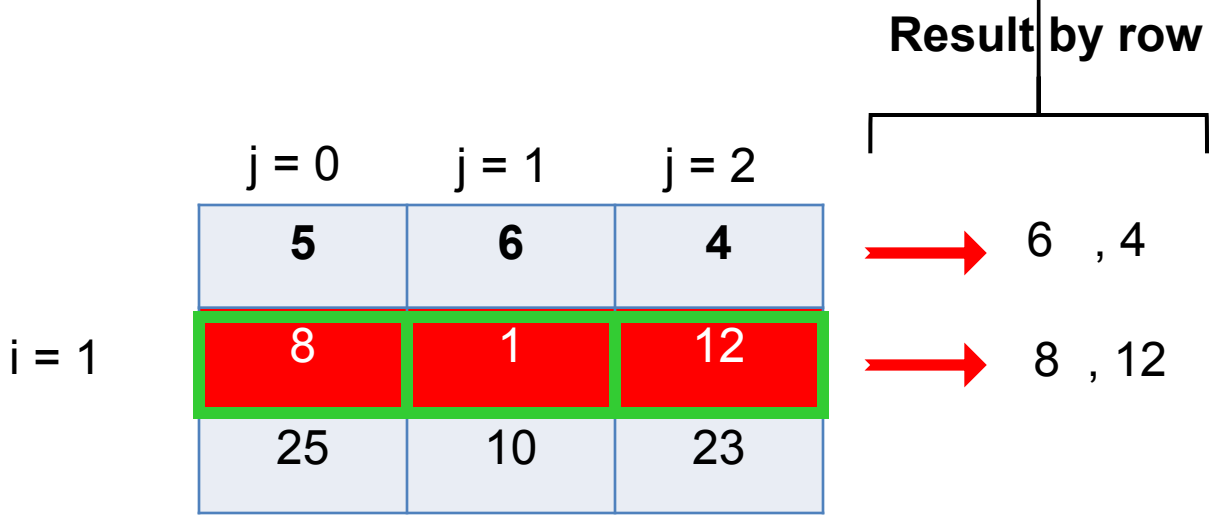
1- Row traversal

- **Example**



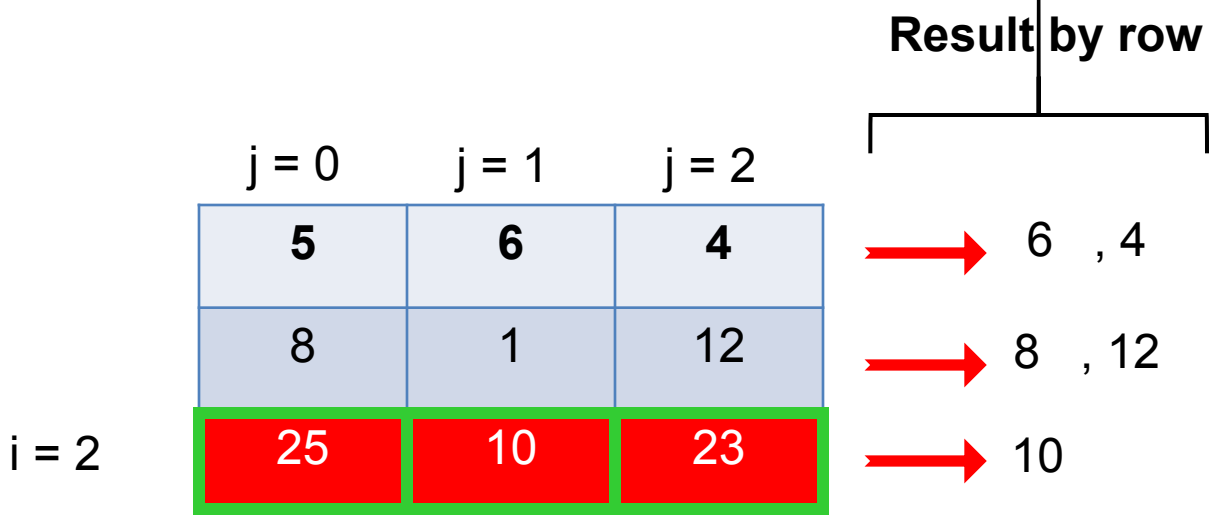
1- Row traversal

- **Example**



1- Row traversal

- **Example**



Final result : 6, 4, 8, 12, 10

1- Row traversal

Example : Sum by Row

Find the sum of each individual row in a matrix :

1- Row traversal

- **Exercise**
- Unroll the following program
- What does the program

```
#include <stdio.h>
int main() {
    int A[3][2] = { {3,5}, {2,4}, {7,2} };
    int xt = 0;
    for (int i = 0; i < 3; i++)
    {
        int x = 0;
        for (int j = 0; j < 2; j++)
            x += A[i][j];
        printf("Row %d: %d\n", i + 1, x);
        xt += x;
    }
    printf("\nxt: %d\n", xt);
    return 0;}
```

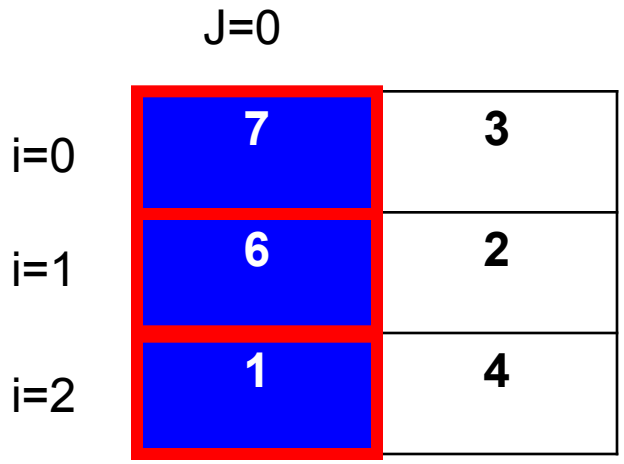
2- Column traversal

Column traversal refers to **visit elements in a matrix column by column**.

7	**3**
6	**2**
1	**4**

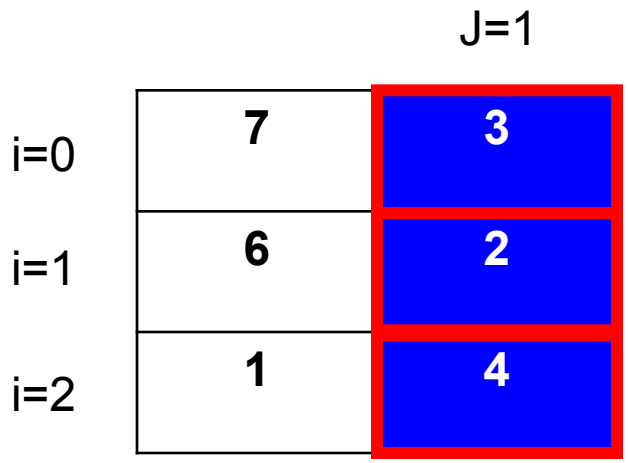
2- Column traversal

Column traversal refers to **visit elements in a matrix column by column**.



2- Column traversal

Column traversal refers to **visit elements in a matrix column by column**.



3- Fixed Row Traversal

Here we keep the row index constant and traverse along the columns.

Second row : i=1

0	5	30
1	12	7
29	0	13

Example. Sum by Fixed Row

To find the sum of **row number 2** in a matrix, we **fix the row** i= 1 and **traverse through the columns**.

4- Fixed column Traversal

Here we keep the **column** index constant and traverse along the row.

Third column:
j=2

0	5	30
1	12	7
29	0	13

Example. Sum by Column

To find the sum of **column number 3** in a matrix, we **fix the column j=2** and **traverse through the rows**.

5- Partial traversal

- A **partial traversal** means only visiting a portion of the elements or performing operations on a subset of data rather than traversing the entire dataset.

12	**0**	**34**	**34**	**3**	**1**
3	9	3	100	44	2
6	7	55	3	22	3
7	6	5	3	39	4
8	5	120	3	20	5

5- Partial traversal

- a **partial traversal** means only visiting a portion of the elements or performing operations on a subset of data rather than traversing the entire dataset.

			j=2		j=4	
	12	**0**	**34**	**34**	**3**	**1**
i=1	3	9	3	100	44	2
	6	7	55	3	22	3
i=3	7	6	5	3	39	4
	8	5	120	3	20	5

```
for (i=P1;i<=P2;i++)
      for (j=Q1;j<=Q2;j++)
```

6- Principal diagonal

0	5	30
1	12	7
29	0	13

```
for (i =0;i<n;i++)
 M[i][i];
```

7- Secondary diagonal

0	5	30
1	12	7
29	0	13

```
for (i =0;i<n;i++)
 M[i][n-1-i];
```

Exercise 1 : (Course)

Given a matrix A of size n*m ($n \leq 20$, $m \leq 30$), write a program which :

1- Reads (fills) the values of the matrix A with two different methods (by row and by column).
2- Displays the maximum and the average of all the elements of A.
3- Construct the matrix B which represents the transpose of the matrix A.
4- Displays the sum of each column of B.
5- Displays matrix A line by line, and matrix B column by column.

Exo (on 0.5 point)

- Ecrire un programme qui affiche la somme des elements pairs des lignes paires

Write a program that displays the sum of the even elements in even rows.