COURSE CODE: CSE 105 & 106
COURSE TITLE: Structured Programming Language

Prepared by- ***Lec Rashid***

Department of CSE,MIST

Why Multidimensional Array is Necessary?

- **Suppose, I want to store the Ct 1 marks of 60 students of both the classes.**
- How many variables do I have to declare?

- We can **solve** it by declaring 2 arrays of 60 students.

  Example – float ct_Marks_EECE[60], ct_Marks_CSE[60];

- Again the same “remembering variable names” problem arises. We can solve this problem by declaring a two dimensional array

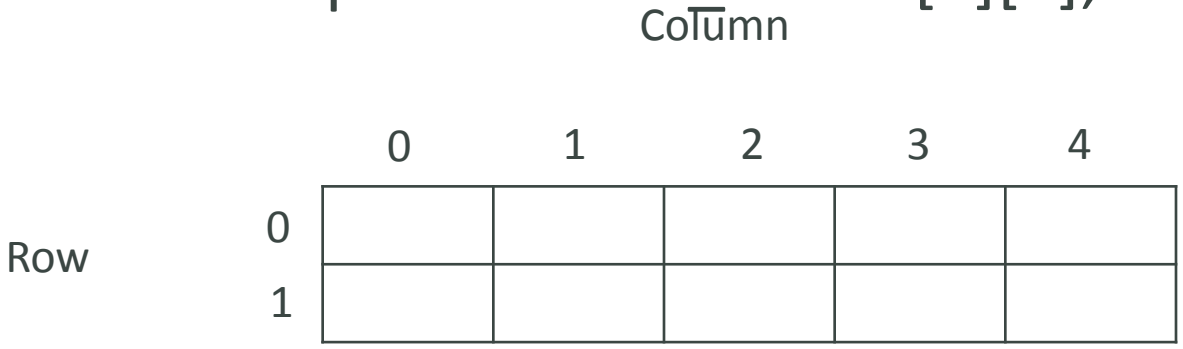
Multidimensional Array

- Two dimensional array declaration:

  Data_Type

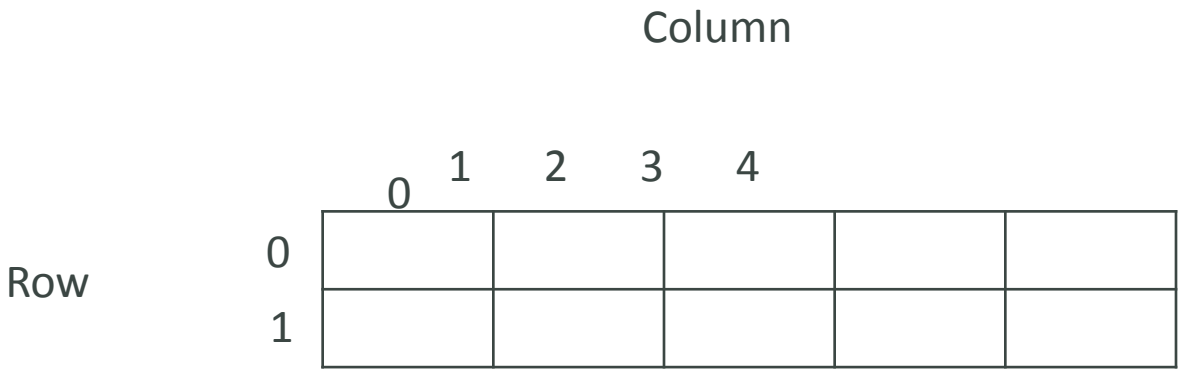
  Array_Name[Row_Size][Column_Size];

  Example: float CT_Marks[2][5];



Multidimensional Array Initialization

- A two dimensional array can be thought as a table with rows and columns
- float ct_Marks[2][5]; - This array can be thought as a table with 2 rows and 5 columns.



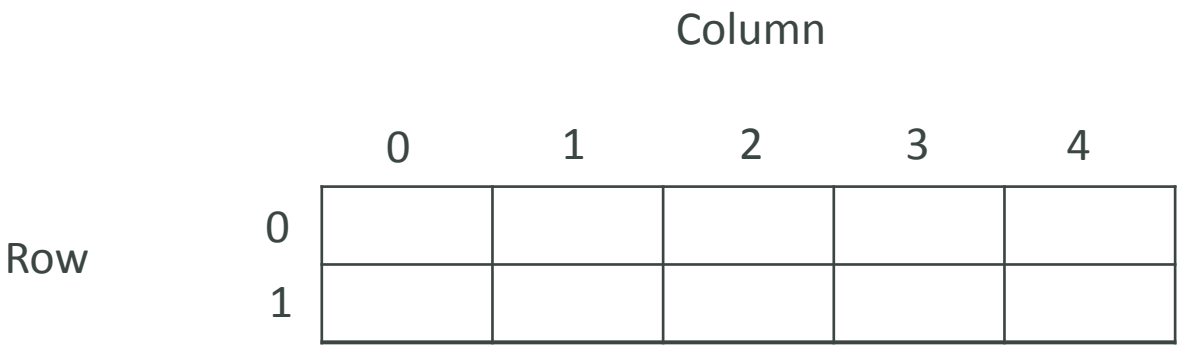
Multidimensional Array Initialization

Process 1 : float ct_Marks[2][5] = {5,10,15,20,25, 5,10,15,20,25}

Process 2 : float ct_Marks[2][5] = {5,10}

Process 3 : float ct_Marks[2][5] = {{5,10,15,20,25}, {5,10,15,20,25}}

Process 4 : float ct_Marks[2][5] = {{5,10,15}, {5,10,15,20,25}}



Multidimensional Array Example 1

```
int main(){
   float ct_Marks[2][10] = {5,10,15,20,25, 5,10,15,20,25};
   for(int row=0; row<2; row++){
    for(int col=0; col<5;col++){
        printf("%.1f  ",ct_Marks[row][col]);
    }
    printf("\n");
   }}
```

Column

Row	0	1	2	3	4
0	5	10	15	20	25
1	5	10	15	20	25

Multidimensional Array Example 2

```
int main(){
   float ct_Marks[2][10]={5,10};

for(int row=0; row<2; row++){
    for(int col=0; col<5;col++){
        printf("%.1f
        ",ct_Marks[row][col]);
    }
    printf("\n");
} }
```

Column

Row	0	1	2	3	4
0	5	10	0	0	0
1	0	0	0	0	0

Multidimensional Array Example 3

```
int main(){
   float ct_Marks[2][10]= {{5,10,15,20,25}, {5,10,15,20,25}};

   for(int row=0; row<2; row++){
    for(int col=0; col<5;col++){
        printf("%.1f
        ",ct_Marks[row][col]);
    }
    printf("\n");
} }
```

Column

Row

	0	1	2	3	4
0	5	10	15	20	25
1	5	10	15	20	25

Multidimensional Array Example 4

```
int main(){
   float ct_Marks[2][10]= {{5,10,15}, {5,10,15,20,25}};

   for(int row=0; row<2; row++){
    for(int col=0; col<5;col++){
        printf("%.1f
        ",ct_Marks[row][col]);
    }
    printf("\n");
} }
```

Column

Row	0	1	2	3	4
0	5	10	15	0	0
1	5	10	15	20	25

Multidimensional Array Example 5

```
int main(){
   float ct_Marks[2][10]={};

   for(int row=0; row<2; row++){
    for(int col=0; col<5;col++){
        printf("%.1f
        ",ct_Marks[row][col]);
    }
    printf("\n");
} }
```

Column

Row	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0

Multidimensional Array Example 6

```
int main(){
   float ct_Marks[2][10];
   ct_Marks[0][0] = 5; ct_Marks[0][1] = 15; ct_Marks[0][2] = 10;

for(int row=0; row<2; row++){
    for(int col=0; col<5;col++){
        printf("%.1f    ",ct_Marks[row][col]);
        Row

    }
    printf("\n");
   }
}
```

Column

Row	0	1	2	3	4
0	**5**	**15**	**10**		
1					

Multidimensional Array Example 7

- Take **input** of the number of **rows** and number of **columns** in the array, Take **input** of the array **element** and **print** them

```
int main(){
  int TotalRow,TotalCol;
  scanf("%d%d",&TotalRow,&TotalCol);
  int arr[TotalRow][TotalCol];
  for(int
   row=0;row<TotalRow;row++){
   for(int col=0; col<TotalCol;col++){
       scanf("%d",&arr[row][col]);
   }
  }
                    for(int
     row=0;row<TotalRow;row++){
    for(int col=0; col<TotalCol;col++){
        printf("%d ",arr[row][col]);
    }
    printf("\n");
   }
}
```

Multidimensional Array Example 8

- 2D Matrix Addition

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$B = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

$$A+B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1+9 & 2+8 & 3+7 \\ 4+6 & 5+5 & 6+4 \\ 7+3 & 8+2 & 9+1 \end{bmatrix}$$

$$= \begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

Multidimensional Array Example 8(Matrix Addition)

- 2D Matrix Addition

A =

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

B =

	0	1	2
0	9	8	7
1	6	5	4
2	3	2	1

A+B =

0	1	2
	2	
	5	
	8	

+

0	1	2
9	8	7
6	5	4
3	2	1

=

$$\begin{bmatrix} 1+9 & 2+8 & 3+7 \\ 4+6 & 5+5 & 6+4 \\ 7+3 & 8+2 & 9+1 \end{bmatrix}$$

=

	0	1	2
0	10	10	10
1	10	10	10
2	10	10	10

Multidimensional Array Example 8(Matrix Addition)

- **First we will take input of the two arrays:**

```
int main(){
   int a[3][3],b[3][3],sum[3][3];
   for(int row=0;row<3;row++){
       for(int col=0;col<3; col++){
          scanf("%d",&a[row][col]);
       }
   }
   for(int row=0;row<3;row++){
       for(int col=0;col<3; col++){
          scanf("%d",&b[row][col]);
       }
   }
```

A =

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

B =

	0	1	2
0	9	8	7
1	6	5	4
2	3	2	1

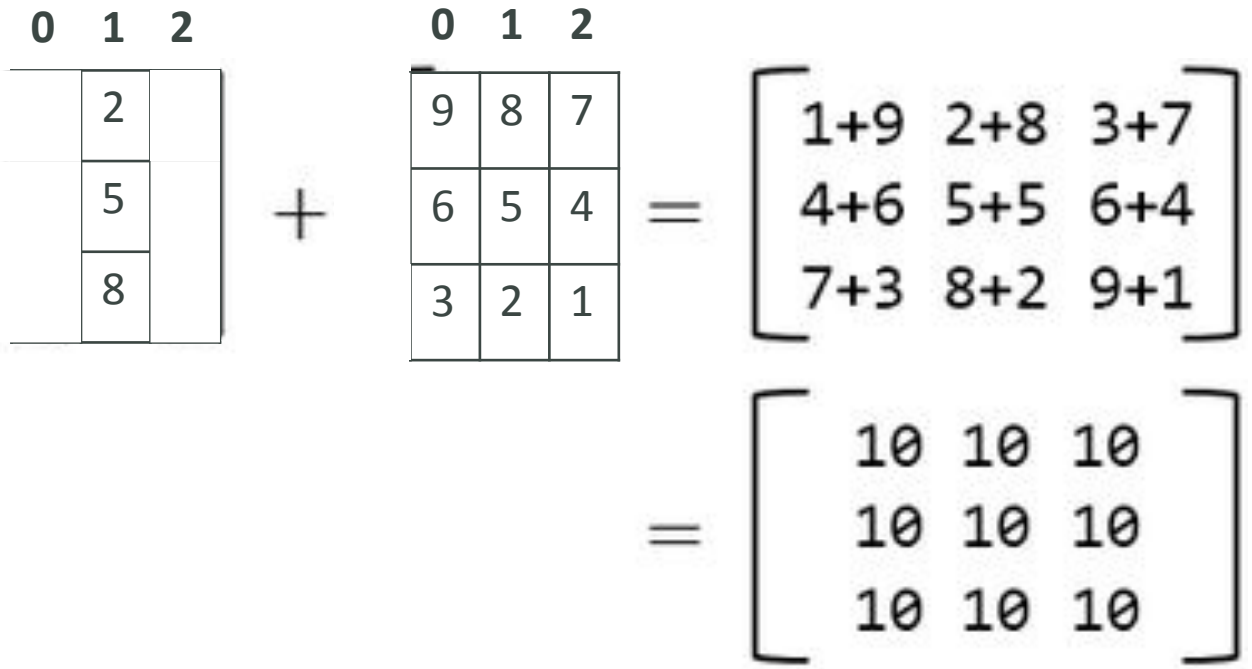
Multidimensional Array Example 8(Matrix Addition)

- **Then we willperform the addition operation**

```
for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){

  sum[row][col]=a[row][col]+b[row][col];
        printf("%d ",sum[row][col]);
      }
      printf("\n");
  }
```

A+B =



Multidimensional Array Example 9

- **Write a program in C to find the sum of all elements of the array.**

```
int main(){
  int a[3][3],sum=0;
  for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         scanf("%d",&a[row][col]);
      }
  }
for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         sum = sum+a[row][col];
         }
  }
  printf("sum: %d ",sum);
}
```

A =

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Tasks

1. Write a program that takes an mxn matrix as input and print the sum of each row of the matrix.
2. Write a program that takes an mxn matrix as input and print the sum of each column of the matrix.
3. Write a program that takes an nXn matrix as input and print the diagonal value of the matrix
4. **C program to find upper triangular matrix**
5. **C program to find lower triangular matrix**
6. Write a program that takes an nxn matrix as input, transpose the matrix and print the matrix.

Task 1

- **Write a program that takes an mxn matrix as input and print the sum of each row of the matrix.**

```
int main(){
  int a[3][3],sum=0;

  for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         scanf("%d",&a[row][col]);
      }
  }
```

```
for(int
      row=0;row<3;row++){
      sum=0;
      for(int col=0;col<3; col++){
         sum=sum+a[row][col];
      }
      printf("sum of row %d is:%d\n",row+1,sum);
  }
}
```

Task 2

- **Write a program that takes an mxn matrix as input and print the sum of each column of the matrix.**

```
int main(){
  int a[3][3],sum=0;

  for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         scanf("%d",&a[row][col]);
      }
  }

for(int col=0;col<3;col++){
      sum=0;
      for(int row=0;row<3;
         row++){
         sum=sum+a[row][col];
      }
      printf("sum of column %d is:%d\n",col+1,sum);
  }
```

Task 3

- **Write a program that takes an nXn matrix as input and print the diagonal value of the matrix**

```
int main(){
  int a[3][3];
  for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         scanf("%d",&a[row][col]);
      }
  }
```

```
for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         if(row==col){
         printf("%d ",a[row][col]);
         }
      }
      printf("\n");
  }
}
```

A =

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Task 4

- **C program to find upper triangular matrix -** To check whether a matrix is upper triangular or not, we need to check whether all elements below main diagonal are zero or not.

```
int main(){
  int a[3][3],flag=0;
  for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
          scanf("%d",&a[row][col]);
      }
  }
```

```
for(int row=0;row<3;row++){
      for(int col=0;col<row; col++){
          if(a[row][col]!=0){
              flag=1;
          }
      }
  }
if(flag==0){
   printf("The matrix is upper triangular");
  }
  else{
   printf("The matrix is not upper triangular");
  }
```

A =

	0	1	2
0	1	2	3
1	0	5	6
2	0	0	9

Task 5

- **C program to find lower triangular matrix -** To check whether a matrix is upper triangular or not, we need to check whether all elements above main diagonal are zero or not.

```
int main(){
  int a[3][3],flag=0;
  for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
          scanf("%d",&a[row][col]);
      }
  }
```

```
for(int row=0;row<3;row++){
      for(int col=row+1;col<3; col++){
          if(a[row][col]!=0){
              flag=1;
          }
      }
  }
  if(flag==0){
   printf("The matrix is lower triangular");
  }
  else{
   printf("The matrix is not lower triangular");
  }
```

A =

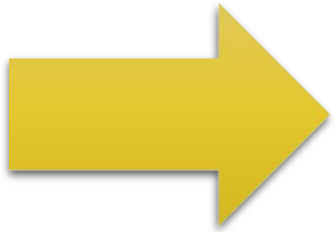
	0	1	2
0	1	0	0
1	4	5	0
2	7	8	9

Task 6

- **Write a program that takes an nxn matrix as input, transpose the matrix and print the matrix.**

```
int main(){
  int a[3][3],b[3][3],sum=0;
  for(int
  row=0;row<3;row++){
      for(int col=0;col<3; col++){
         scanf("%d",&a[row][col]); }
  }
for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         b[col][row] = a[row][col];
         }
```

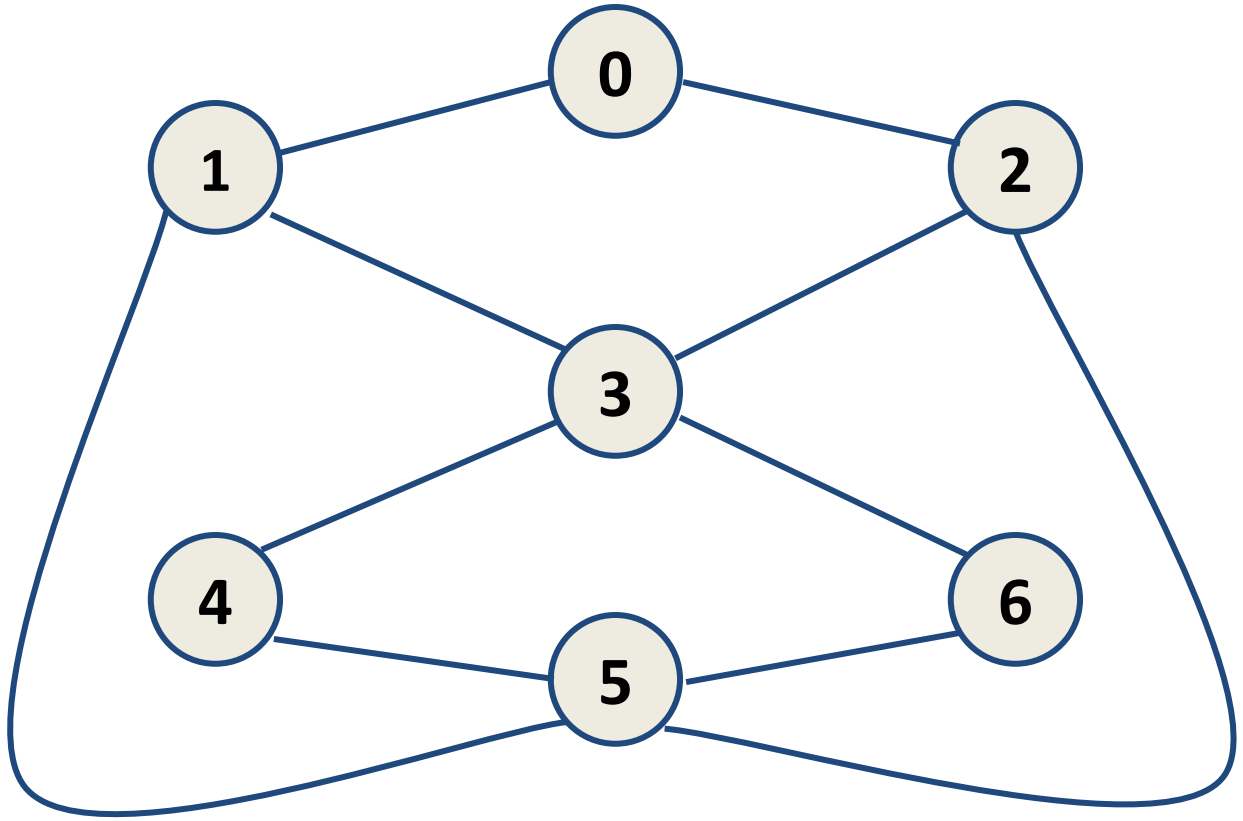
1	2	3	4
9	10	11	12
5	6	7	8
13	14	15	16



1	9	5	13
2	10	6	14
3	11	7	15
4	12	8	16

```
for(int row=0;row<3;row++){
      for(int col=0;col<3; col++){
         printf("%d ",b[row][col]);
        }
        printf("\n");
  }
```

Graph - Adjacency Matrix



	0	1	2	3	4	5	6
0	0	1	1	0	0	0	0
1	1	0	0	1	0	1	0
2	1	0	0	1	0	1	0
3	0	1	1	0	1	0	1
4	0	0	0	1	0	1	0
5	0	1	1	0	1	0	1
6	0	0	0	1	0	1	0

Graph - Adjacency Matrix and Bipartite Graph Problem

You are given a task to color an undirected graph with black and white color. ‘1’ represents Black and ‘0’ represents White. Now you are given ‘n’ number of nodes with ‘e’ numbers of edges connecting them. You are tasked to color the graph such that no adjacent nodes has the same color. You have to find out that if it’s possible to color that graph or not.

Sample Input:

```
7
10
0 1
0 2
1 3
2 3
3 4
4 5
5 6
3 6
1 5
2 5
```

Sample Output:

Possible


Thank
you!