Memory Representation

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

	Column 0	Column 1	Column 2	Column 3
Row 0	0	1	2	3
Row 1	4	5	6	7
Row 2	8	9	10	11

To access value 1 we write a[0][1]; To access value 11 we write a[2][3]

Basic Syntax

DataType arrayName [row][column];

How to declare the 2D arrays:-

int a[3][4];

In the declaration above row=3 and column=4

OR

```
int a[3][4] = {
  {0, 1, 2, 3} ,  /* row 0 */
  {4, 5, 6, 7} ,  /* row 1 */
  {8, 9, 10, 11}  /* row 2 */
};
```

OR

The above statement is same as:

int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};

Declaring and accessing the elements of a two-dimensional array:

```
int main()
{
    int A[100][100], i, j, rows, columns;
    printf("Number of rows: ");
    scanf("%d",&rows);
    printf("Number of columns: ");
    scanf("%d",&columns);
    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
        {
            printf("A[%d][%d]: ",i, j);
            scanf("%d",&A[i][j]);
        }
    }
    printf("Values in array A:\n");
    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
        {
            printf("%10d ",A[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Problems:

1. i) Declare a 2 dimensional array of row size 3 and column size 3.
   ii) Fill the 2 dimensional array with values from the user using nested loops.
   iii) Search for a user given value in the 2D array by using if-else and nested loops. If the value is not present then print "Value not found", print "Value found" if present.

2. Ask user for number of rows and number of columns. Based on input, declare two 2-dimensional arrays M and N. Now implement the following tasks:

   i) Take input for both arrays
   ii) Output the arrays as form of matrix
   iii) Calculate the sum and store the sum in another 2D array of same dimension. Display the sum.

   Sample run:

Row: 2 Column: 3	Enter first array elements: 2 3 1 10 4 6	Enter second array elements: 7 2 4 6 8 4	First array: 2 3 1 10 4 6 Second array: 7 2 4 6 8 4	Sum: 9 5 5 16 12 10

3. Write a program that computes the sum of a specific column (provide by user as input) in a 2D array.

Sample Output:

Enter number of rows: **3**
Enter number of columns: **4**

Enter elements: **2 3 6 7**
4 **8 1 5**
9 **0 7 2**

Enter which column to find sum of: **2**
Sum of column 2: **11**

4. Take input of a matrix of MxN dimension, where M and N are user inputs. Now display the transpose of it. You can get the transpose matrix by interchanging row and column of the original matrix. See the following example:

Original matrix	Transpose matrix
4 6 2 1 3 8	4 1 6 3 2 8