**Matrix OR 2D Array in CPP –**

Matrix/2-D Array Programs -

1 - Basic Input/Output of 2D Array.

2 - Print Sum of all elements in the matrix -

3 - check for the target element that it is present in the matrix or not, if present then print Yes.

4 - Find the absolute difference of both diagonal elements of a Matrix

5 - Find the sum of only boundary elements in the matrix -

6 - Transpose of a Matrix -

7 - Rotate the Matrix by 180 degrees.

8 - Rotate the array by 90 CLW.

9 - Rotate the array by 90 ACLW.

**Matrix OR 2D Array in CPP –**

#include <bits/stdc++.h>

// #include <iostream>

// #include<algorithm>

// #include<climits>

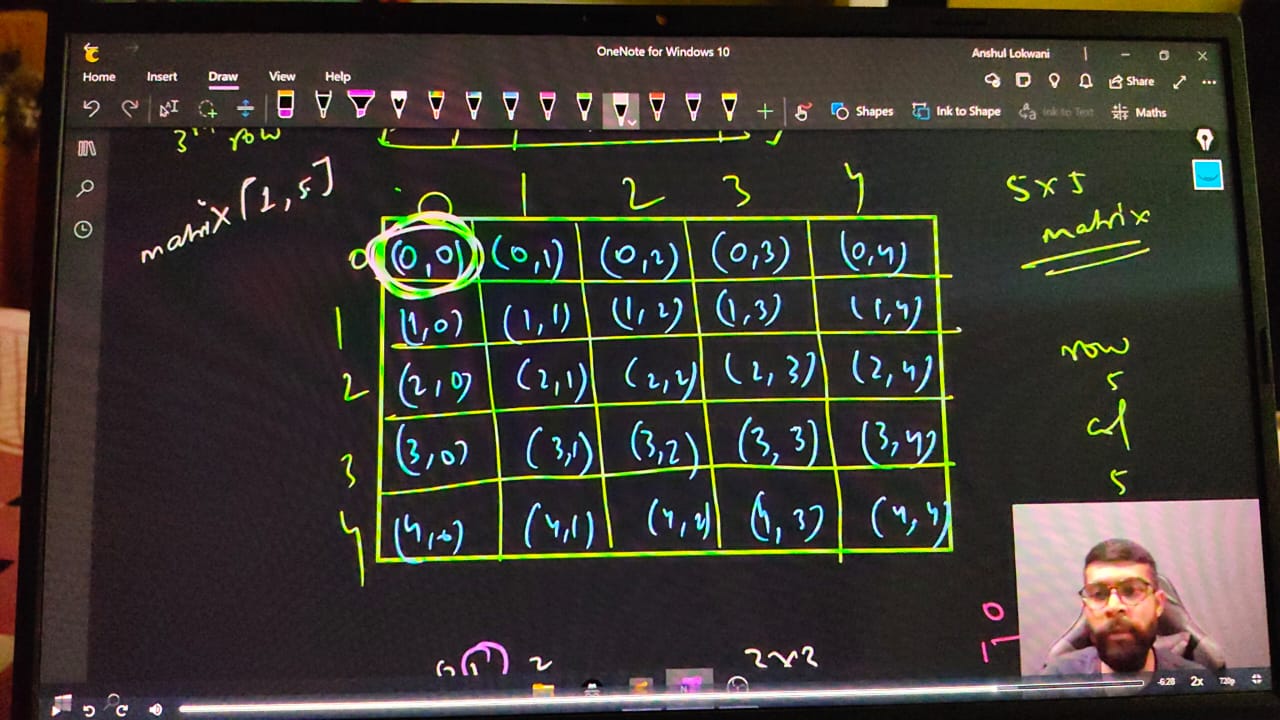
// #include<string>

// #include<cctype>

using namespace std;

int main()

{



  /\*

**initialize a 2D Array -**

**Standard Syntax -**

    DATATYPE Name ARRAYNAME[No.of Row][No.ofColumn];

    \*/

    // int matrix[5][5];

    // matrix[1][2] = 10;

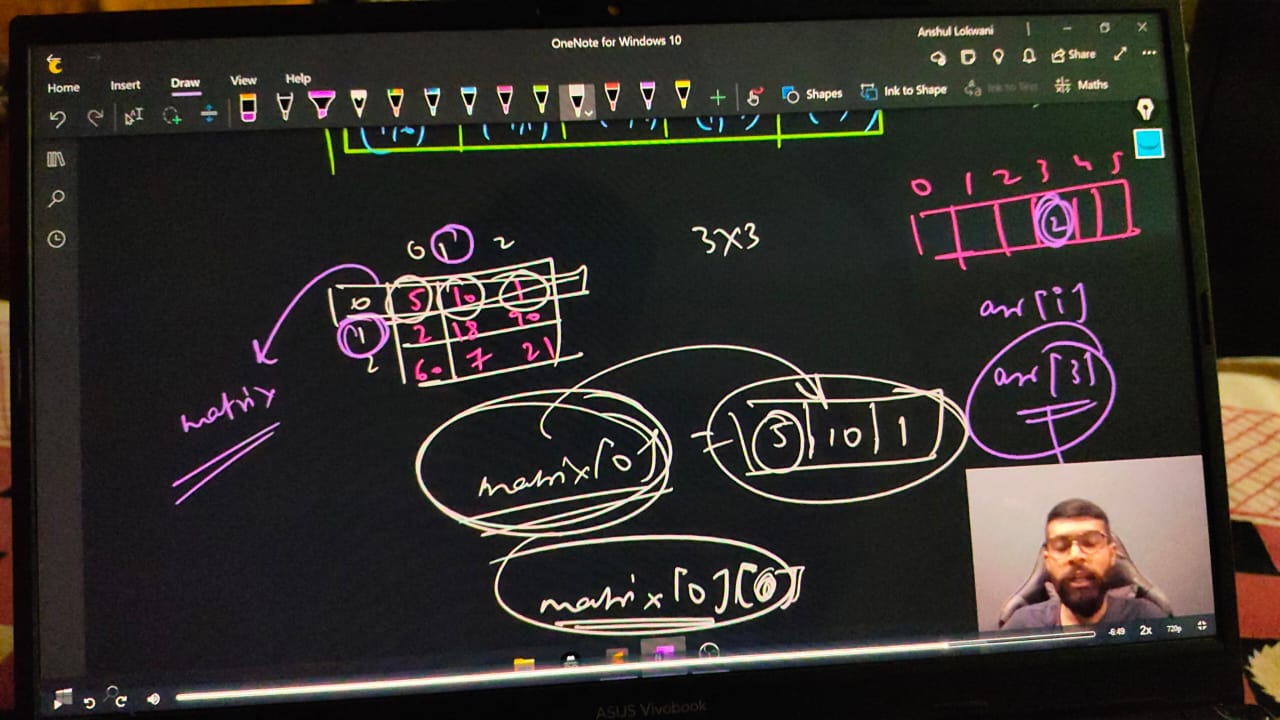
    // cout << matrix[1][2]; // 10

    // cout << endl;

    //    taking input manally for a single value

    // cin>>matrix[1][2];//75

    // cout<<matrix[1][2];//75



**// taking inputs for all the values of matrix -**

    // int n,m;

    // cout<<"Enter the no. of rows and columns respectively"<<endl;

    // cin>>n>>m;

    // int matrix2[n][m];

    // cout<<"Now, enter the elements value"<<endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin>>matrix2[i][j];

    //     }

    // }

**// // Printing the entered matrix -**

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout<<matrix2[i][j]<<" ";

    //     }

    //     cout<<endl;

    // }

    /\*

    Enter the no. of rows and columns respectively

    3 3

    Now, enter the elements value

    1 2 3 4 5 6 7 8 9

    1 2 3

    4 5 6

    7 8 9

    \*/

**// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_**

**// Qun2 - Print SUm of all elements in the matrix -**

    // int n,m;

    // cout<<"Enter the no. of rows and columns respectively"<<endl;

    // cin>>n>>m;

    // int matrix2[n][m];

    // // taking input from user

    // cout<<"Now, enter the elements value"<<endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin>>matrix2[i][j];

    //     }

    // }

    // // Printing the entered matrix -

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout<<matrix2[i][j]<<" ";

    //     }

    //     cout<<endl;

    // }

    //     int sum=0;

    // // Printing the sum of entered matrix -

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         sum+=matrix2[i][j];

    //     }

    // }

    // cout<<"So, the sum of entire matrix is - "<<sum<<endl;

    /\*

Enter the no. of rows and columns respectively

3 3

Now, enter the elements value

1 2 3 4 5 6 7 8 9

1 2 3

4 5 6

7 8 9

So, the sum of entire matrix is - 45

    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// QUn 3 -check for the target element that it is present in the matrix or not, if present then print Yes.**

    // int n, m;

    // cout << "Enter the no. of rows and columns respectively" << endl;

    // cin >> n >> m;

    // int matrix2[n][m];

    // // taking input from user

    // cout << "Now, enter the elements value" << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin >> matrix2[i][j];

    //     }

    // }

    // // Printing the entered matrix -

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout<<matrix2[i][j]<<" ";

    //     }

    //     cout<<endl;

    // }

    // int target;

    // cout << "So, what's the targeted element - " << endl;

    // cin >> target;

    // // Printing the sum of entered matrix -

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         if (matrix2[i][j]==target)

    //         {

    //             cout<<"Yes it's present and the address is: row no-  "<<i<<" & column no-  "<<j<<" having targeted value-  "<<matrix2[i][j];

    //             return 0;

    //         }

    //     }

    // }

    // cout<<"Sorry, Element Not FOund Bruh.."<<endl;

    /\*

    Enter the no. of rows and columns respectively

    3 3

    Now, enter the elements value

    1 2 3 4 5 6 7 8 9

    1 2 3

    4 5 6

    7 8 9

    So, what's the targeted element -

    6

    Yes it's present and the address is: row no-  1 & column no-  2 having targeted value-  6

    Enter the no. of rows and columns respectively

    3 3

    Now, enter the elements value

    1 2 3 4 5 6 7 8 9

    1 2 3

    4 5 6

    7 8 9

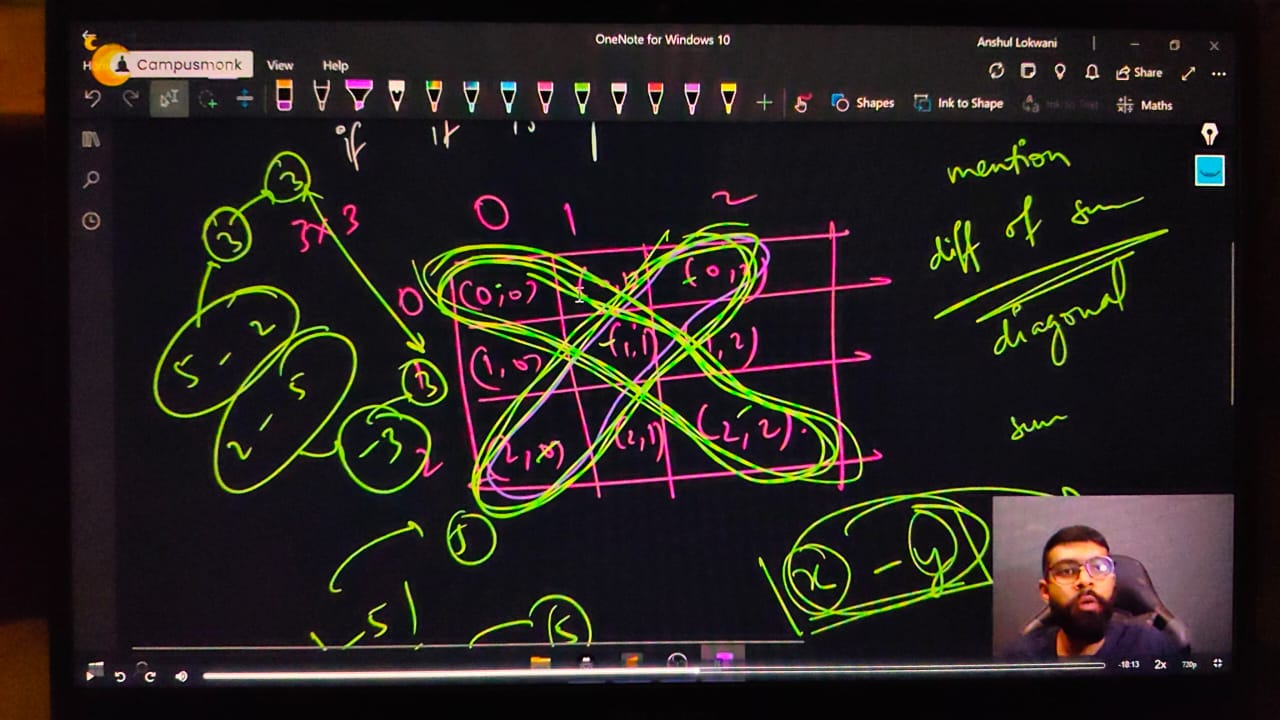
    So, what's the targeted element -

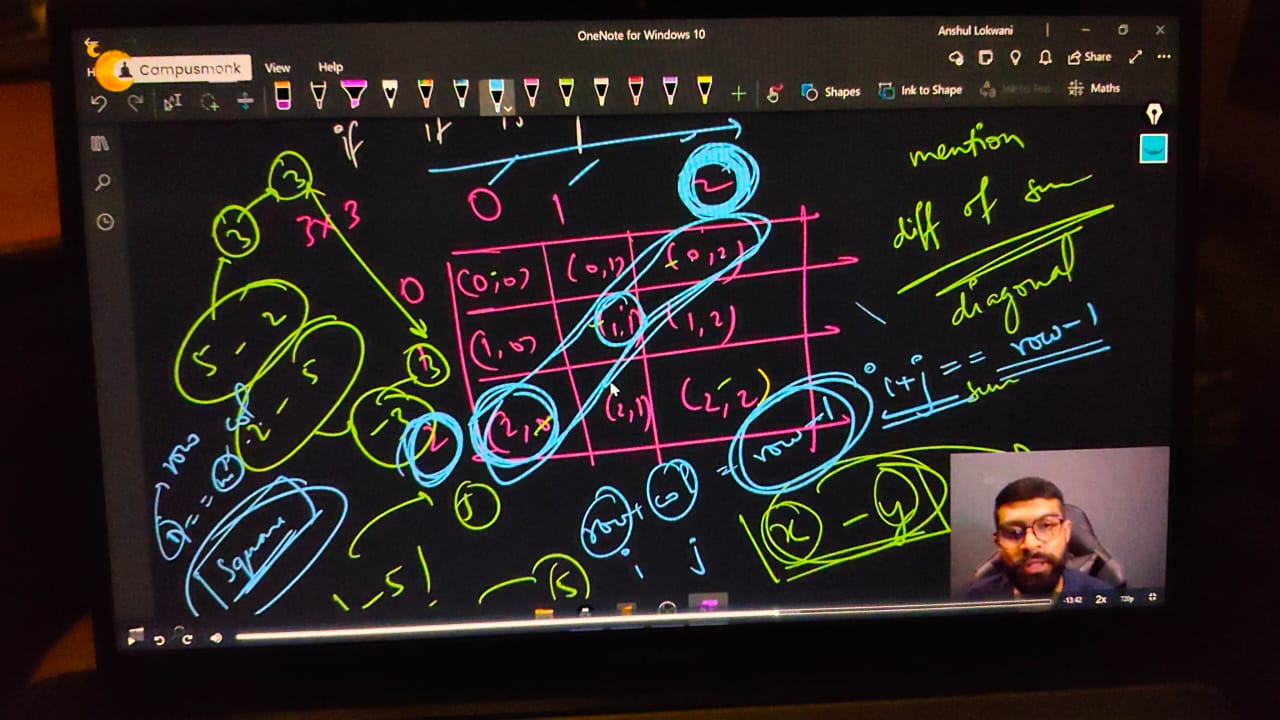
    15

    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// QUn 4 - Find the absolute difference of both diagonal elemenmts of a Matrix -**





    // int rows, colms;

    // cout << "Enter the roes and column va;ues resprectively" << endl;

    // cin >> rows >> colms;

    // int matrix[rows][colms];

    // cout << "Enter the elements of matrix - " << endl;

    // for (int i = 0; i < rows; i++)

    // {

    //     for (int j = 0; j < colms; j++)

    //     {

    //         cin >> matrix[i][j];

    //     }

    // }

    // cout << "So, youe entered matrix is - " << endl;

    // for (int i = 0; i < rows; i++)

    // {

    //     for (int j = 0; j < colms; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    // int SumofDiagonal1 = 0;

    // int SumofDiagonal2 = 0;

    // // for principle diagonal elements sum - its sum of dialoganl1

    // for (int i = 0; i < rows; i++)

    // {

    //     for (int j = 0; j < colms; j++)

    //     {

    //         if (i == j)

    //         {

    //             SumofDiagonal1 += matrix[i][j];

    //         }

    //         if (i + j == rows - 1)

    //         {

    //             SumofDiagonal2 += matrix[i][j];

    //         }

    //     }

    // }

    // // For another dialognal sum -

    // cout << "So the sum of PDE is - " << SumofDiagonal1 << " and another diagonal is  " << SumofDiagonal2 << endl;

    // cout << "So, clearly the difference of both diagonals is -  " << abs(SumofDiagonal1 - SumofDiagonal2) << endl;

    // For the absolute difference using abs functin

    /\*

    Enter the roes and column va;ues resprectively

    3 3

    Enter the elements of matrix -

    1 2 3 4 5 6 7 8 9

    So, youe entered matrix is -

    1 2 3

    4 5 6

    7 8 9

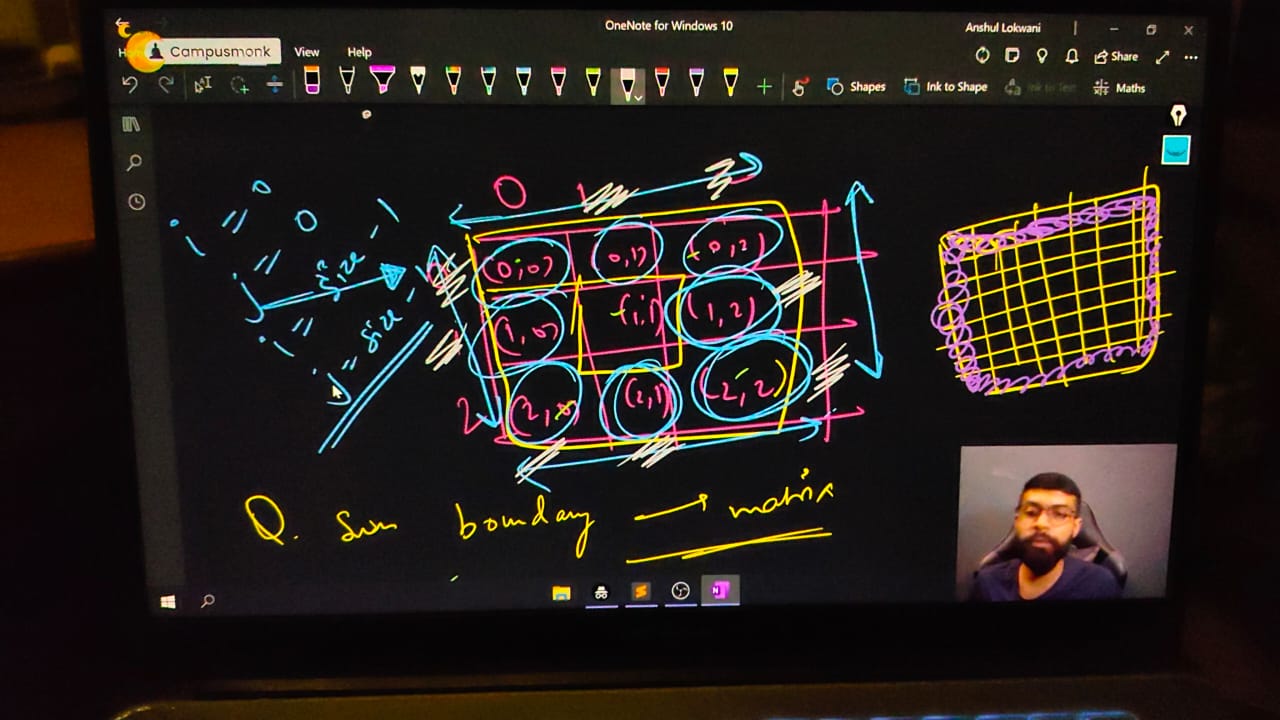
    So the sum of PDE is - 15 and another diagonal is  15

    So, clearly the difference of both diagonals is -  0

    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// Qun 5 - Find the sum of only boundary elements in the matrix –**



    // int n, m;

    // cout << "Enter the no. of rows and columns respectively" << endl;

    // cin >> n >> m;

    // int matrix2[n][m];

    // // taking input from user

    // cout << "Now, enter the elements value" << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin >> matrix2[i][j];

    //     }

    // }

    // // Printing the entered matrix -

    // cout<<"So, your inserted elements in matrix is - "<<endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix2[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    // // For Boundry ELemnets Sum -

    // // 📌 Logic - row & colm is max and min which is 0 and n-1,m-1

    // int boundary\_sum = 0;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         if (i == 0 || j == 0 || i == n - 1 || j == m - 1)

    //         {

    //             boundary\_sum += matrix2[i][j];

    //         }

    //     }

    // }

    // cout<<"So the sum of all boundaries is - "<<boundary\_sum<<endl;

    /\*

    Enter the no. of rows and columns respectively

    3 3

    Now, enter the elements value

    1 2 3 4 5 6 7 8 9

    1 2 3

    4 5 6

    7 8 9

    So the sum of all boundaries is - 40

    Enter the no. of rows and columns respectively

    3 4

    Now, enter the elements value

    12 5 4 10 7 3 6 9

    1 2 5 3 4 7 6 9 8

    So, your inserted elements in matrix is -

    12 5 4 10

    7 3 6 9

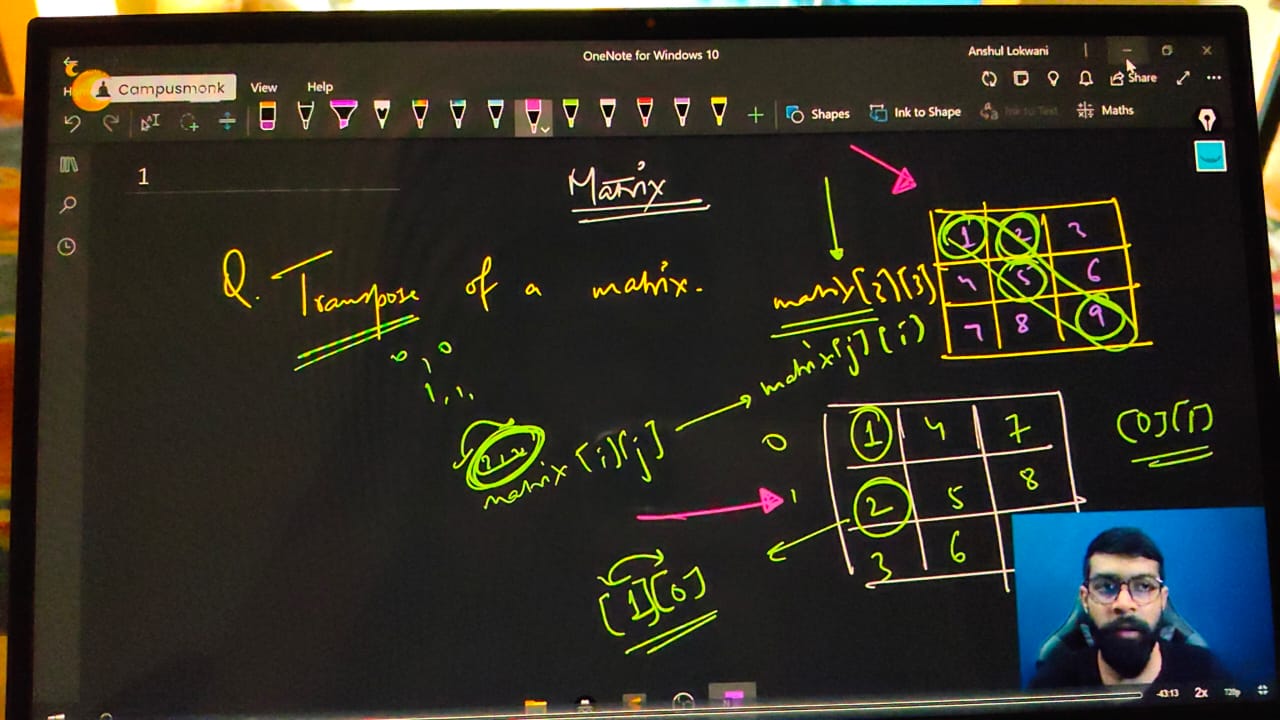
    1 2 5 3

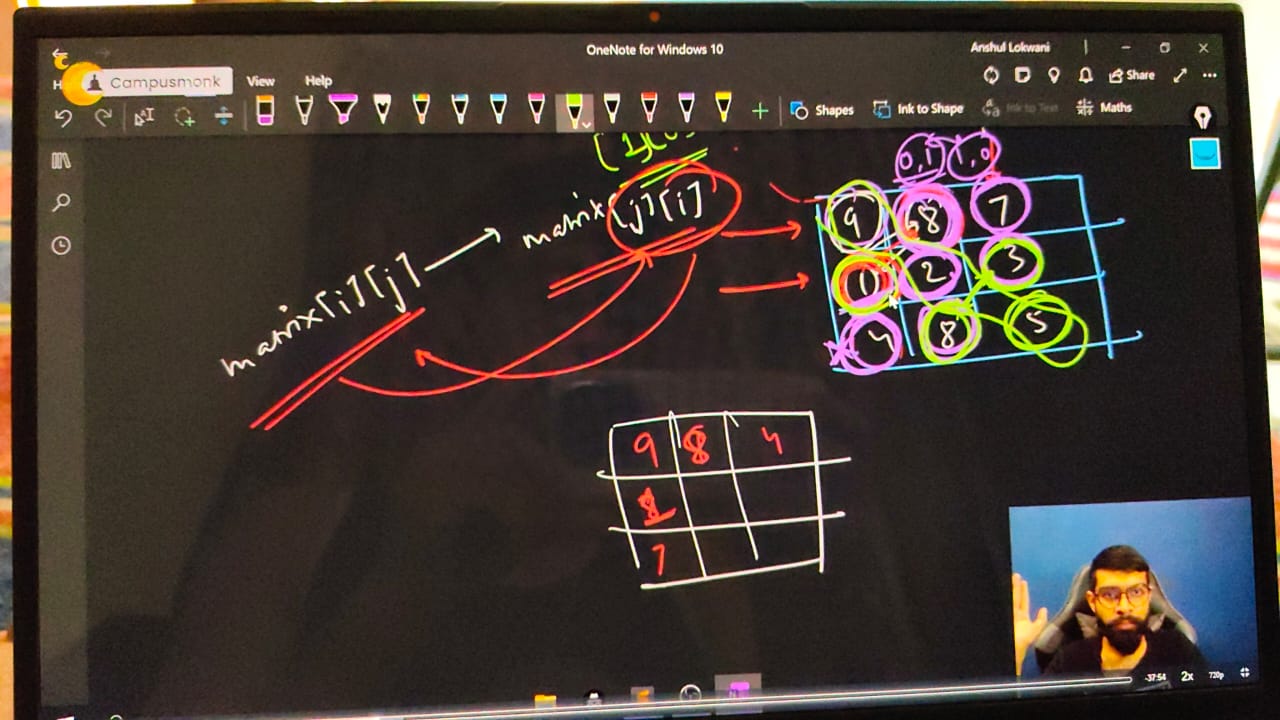
    So the sum of all boundaries is - 58

    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// Qun 6 - Transpose of a Matrix –**





    // int n, m;

    // cout << "ENter the no. of Roews and Columns you want" << endl;

    // cin >> n >> m;

    // int matrix[n][m];

    // cout << "Now insert the matrix elements, mention you favourite Angel numbers " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin >> matrix[i][j];

    //     }

    // }

    // cout<<endl;

    // cout << "So, the entered Array Matrix is - " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    // cout << endl;

    // For Printing the Transposed Matrices -

    /\*

    diagonal ke phle phle hi swap kr rhe h qki uske baad bhi krenge to vapas vvahii matrix aa jayega. That's why used j<i. diagonal ke under under

    Diagonal ke left me - using for (int j = 0; j < i; j++)

    Diagonal ke right me - using for(int j=i; j<n; j++)

    simple diagonal ki ek side ko dusrti side se swap krna he. agr dono side se swap krenge to vapas vahi matrix milega

    \*/

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < i; j++)

    //     {

    //         swap(matrix[i][j], matrix[j][i]);

    //     }

    // }

    // cout << "And the Transposed Matrices is " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    /\*

    ENter the no. of Roews and Columns you want

    3 3

    Now insert the matrix elements, mention you favourite Angel numbers

    1 2 3 4 5 6 7 8 9

    So, the entered Array Matrix is -

    1 2 3

    4 5 6

    7 8 9

    So, the Transposed Matrices is

    1 4 7

    2 5 8

    3 6 9

    ENter the no. of Roews and Columns you want

    3 3

    Now insert the matrix elements, mention you favourite Angel numbers

    15 10 12 23 56 45 90 87 30

    So, the entered Array Matrix is -

    15 10 12

    23 56 45

    90 87 30

    And the Transposed Matrices is

    15 23 90

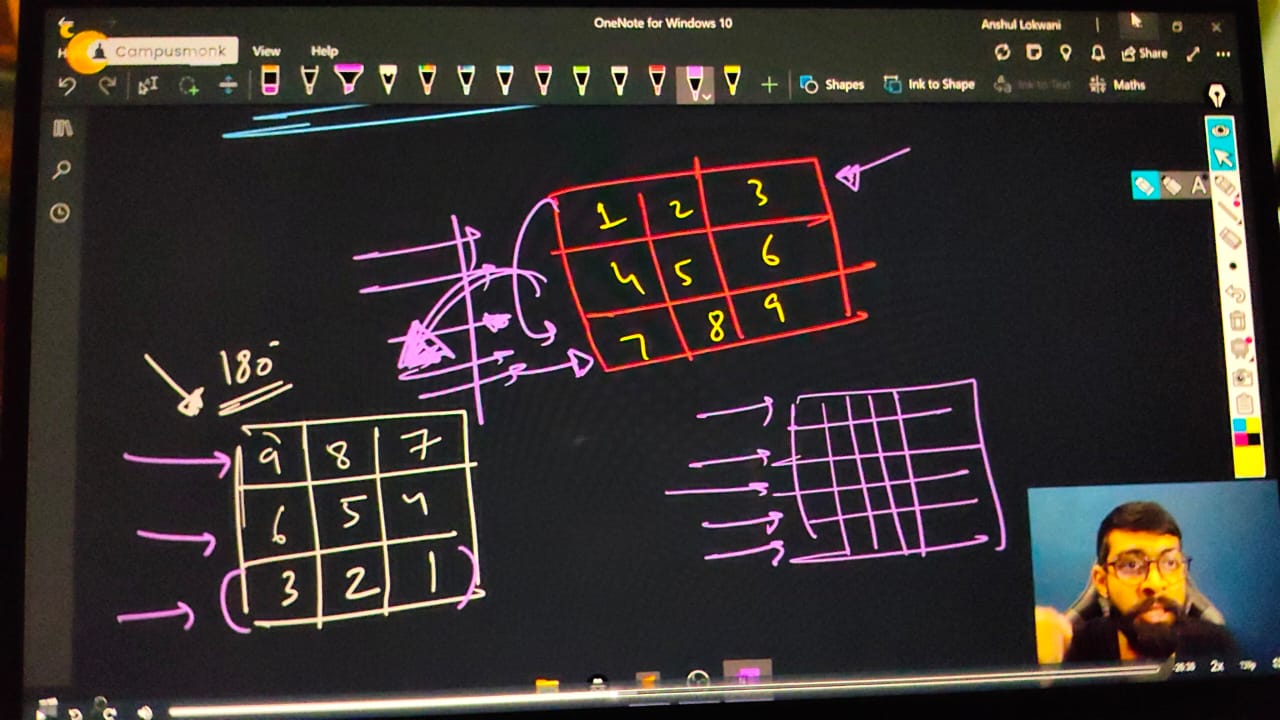
    10 56 87

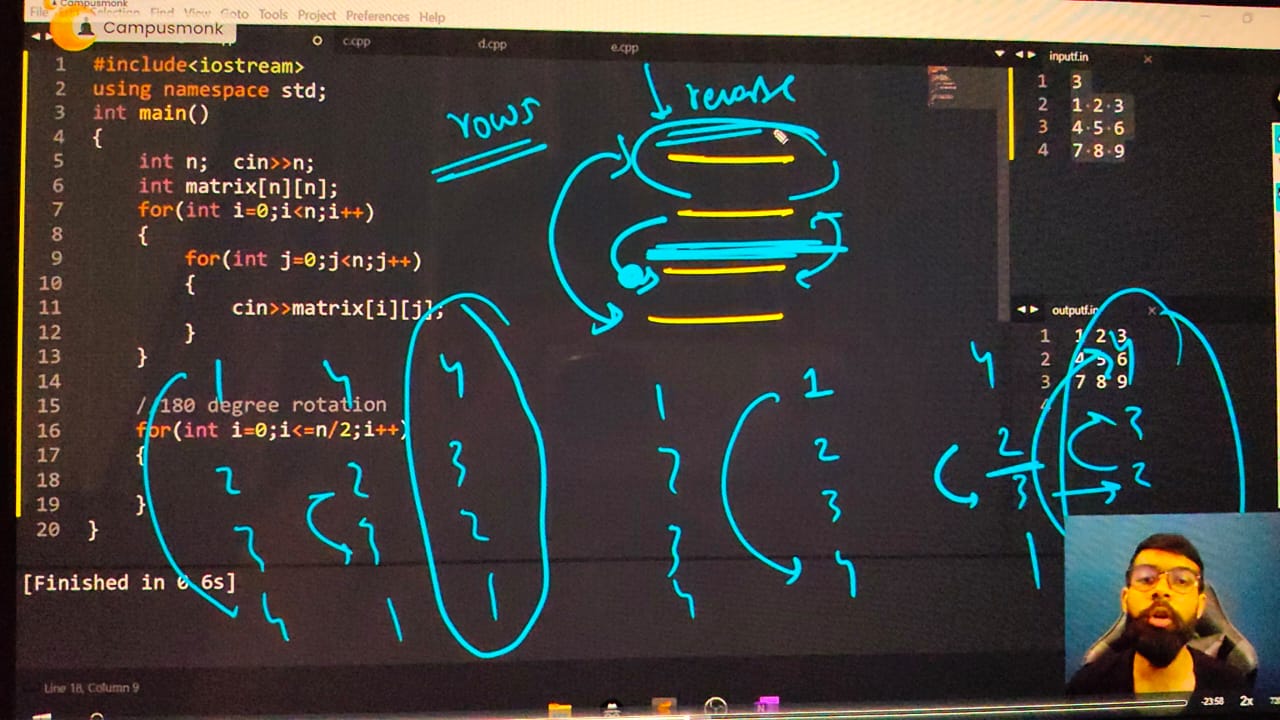
    12 45 30

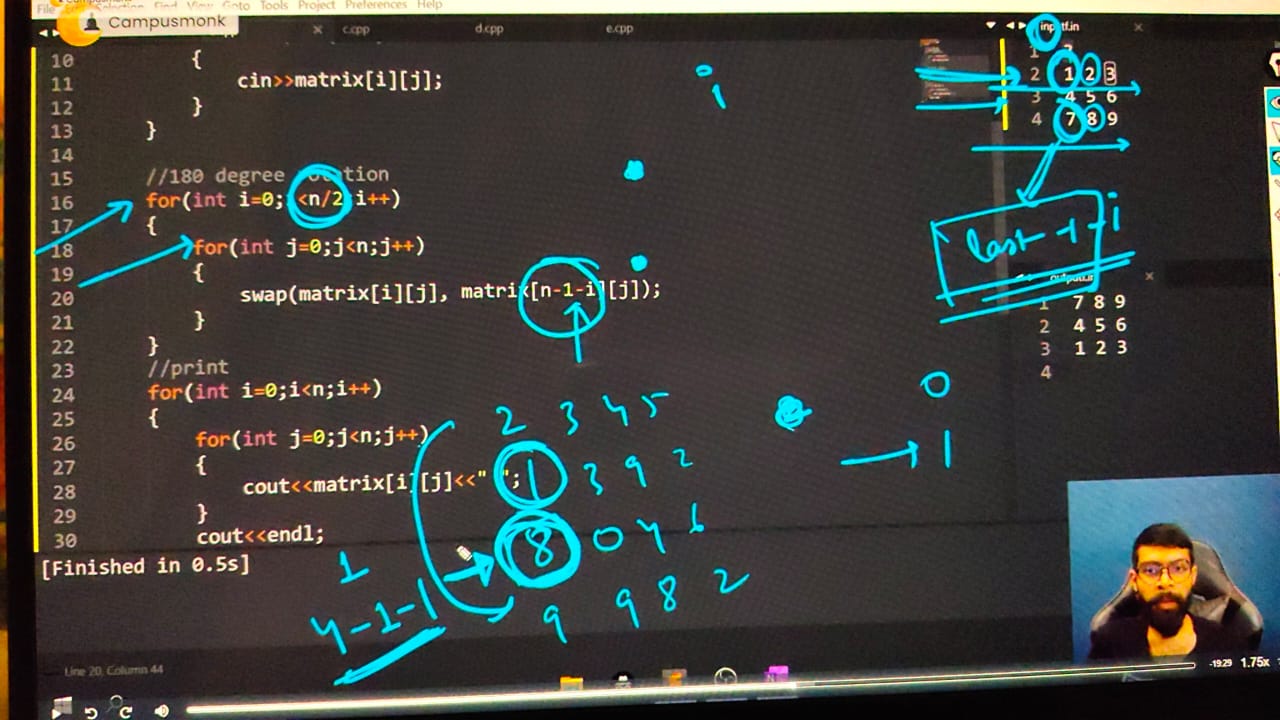
    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// Qun 7 - Rotate the Matrix by 180 degree.**







    // int n, m;

    // cout << "ENter the no. of Roews and Columns you want" << endl;

    // cin >> n >> m;

    // int matrix[n][m];

    // cout << "Now insert the matrix elements, mention you favourite Angel numbers " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin >> matrix[i][j];

    //     }

    // }

    // cout << endl;

    // cout << "So, the entered Array Matrix is - " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    // // For 180 degree rotation of AMatrix

    // /\*

    // 📌 Logic - 180 rotate krne k liye samajh aaya  ki 1st row ko last row se swap krna he and 2nd rowe ko last 2nd se. or 1st and 3rd rows ke elements ko reverse krna he.

    //  - phle 1st row ko last se swap kiya phr 2nd row ko 2nd last se kiya to yh to pura reverse ho gya matrix which is 180 degree rotation

    //  - to agr n rowws he to n/2 times swap hoga - for reverse all rows

    //  - Remeber in matrix - bich waali row ko bich waali row se ya 1st ko last se - we use subsequent row which is alsywas given as -  n(TotalRows)-1-i(roww no)

    //  \*/

    // Row Swap

    // for (int i = 0; i < n / 2; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         swap(matrix[i][j], matrix[n - 1 - i][j]);// n-1-i uses for get the subsequennt row of current row

    //     }

    // }

    // // Reverse bhi krna he elements ko so that could be in 180 deg

    // for (int i = 0; i < n; i++)

    // {

    // // Syntax for reverse - reverse(ARRAYNAME, ARRAYNAME+ARRAYSIZE)

    // // Modified syntax as per trials - reverse(ARRAYNAME+jaha tk reverse krna ho, ARRAYNAME+jaha se reverse krna ho)

    //     reverse(matrix[i], matrix[i]+n);

    // }

    // cout<<"The Matrix after Rotations is - "<<endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    /\*

    ENter the no. of Roews and Columns you want

    3 3

    Now insert the matrix elements, mention you favourite Angel numbers

    1 2 3 4 5 6 7 8 9

    So, the entered Array Matrix is -

    1 2 3

    4 5 6

    7 8 9

    The Matrix after Rotations is -

    9 8 7

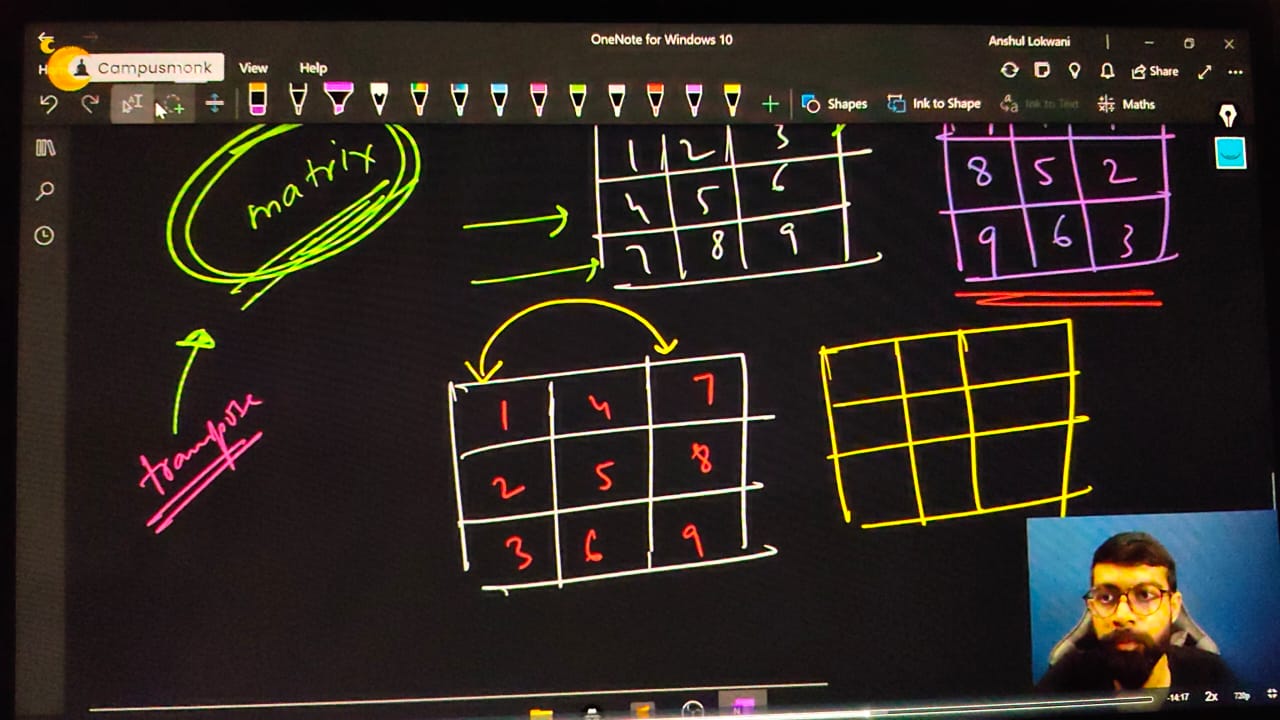
    6 5 4

    3 2 1

    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// Qun 8 - Rotate the array by 90 CLW.**



    // int n, m;

    // cout << "ENter the no. of Roews and Columns you want" << endl;

    // cin >> n >> m;

    // int matrix[n][m];

    // cout << "Now insert the matrix elements, mention you favourite Angel numbers " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin >> matrix[i][j];

    //     }

    // }

    // cout << endl;

    // cout << "So, the entered Array Matrix is - " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    // /\*

    // 📌 Logic - 1) ro to column from last to 1st column

    // OR 2) take traspose and then swap columns

    // \*/

    // // Tanspose Mat -

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = i; j < n; j++)

    //     {

    //         swap(matrix[i][j], matrix[j][i]);

    //     }

    // }

    // // cout<<"The ranspse mat is - "<<endl;

    // // for (int i = 0; i < n; i++)

    // // {

    // //     for (int j = 0; j < m; j++)

    // //     {

    // //         cout << matrix[i][j] << " ";

    // //     }

    // //     cout << endl;

    // // }

    // // Now swap the columns. Column 1 with column 3 and 2nd column with 2nd last column

    // // So for it reverse the columns

    // for (int j = 0; j < m; j++)

    // {

    //     reverse(matrix[j], matrix[j]+n);

    // }

    // OR using -colun swap

    // for (int i = 0; i < n/2; i++)

    // {

    //     for (int j = 0; j < n; j++)

    //     {

    //         swap(matrix[j][i], matrix[j][n-1-i]);

    //     }

    // }

    // cout<<"So The Matrix after 90 degree rotation clw is "<<endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    /\*

    ENter the no. of Roews and Columns you want

    3 3

    Now insert the matrix elements, mention you favourite Angel numbers

    1 2 3 4 5 6 7 8 9

    So, the entered Array Matrix is -

    1 2 3

    4 5 6

    7 8 9

    So The Matrix after 90 degree rotation clw is

    7 4 1

    8 5 2

    9 6 3

    \*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**// Qun 9 - Rotate the array by 90 ACLW.**

    // int n, m;

    // cout << "ENter the no. of Roews and Columns you want" << endl;

    // cin >> n >> m;

    // int matrix[n][m];

    // cout << "Now insert the matrix elements, mention you favourite Angel numbers " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cin >> matrix[i][j];

    //     }

    // }

    // cout << endl;

    // cout << "So, the entered Array Matrix is - " << endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

    // /\*

    // 📌 Logic - 1) row to column from 1st to last in reverse order of elements

    // OR 2) take traspose and then swap Rows

    // \*/

    // // Tanspose Mat -

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = i; j < n; j++)

    //     {

    //         swap(matrix[i][j], matrix[j][i]);

    //     }

    // }

    // // cout << "The Transpose matrix is - " << endl;

    // // for (int i = 0; i < n; i++)

    // // {

    // //     for (int j = 0; j < m; j++)

    // //     {

    // //         cout << matrix[i][j] << " ";

    // //     }

    // //     cout << endl;

    // // }

    // //Now swap the rows to get the matrix rotation by 90 degress in ACW Manner -

    // for (int i = 0; i < n / 2; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         swap(matrix[i][j], matrix[n - 1 - i][j]);// n-1-i uses for get the subsequennt row of current row

    //     }

    // }

    // cout<<"and, finally 90 deg rotated Matrix by ACW"<<endl;

    // for (int i = 0; i < n; i++)

    // {

    //     for (int j = 0; j < m; j++)

    //     {

    //         cout << matrix[i][j] << " ";

    //     }

    //     cout << endl;

    // }

/\*

ENter the no. of Roews and Columns you want

3 3

Now insert the matrix elements, mention you favourite Angel numbers

1 2 3 4 5 6 7 8 9

So, the entered Array Matrix is -

1 2 3

4 5 6

7 8 9

and, finally 90 deg rotated Matrix by ACW

3 6 9

2 5 8

1 4 7

\*/

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

}