

```
/*
```

Given a M x N matrix mat[], the task is to count the number of cells which have the sum of its adjacent cells equal to a prime number. For a cell x[i][j], only x[i+1][j], x[i-1][j], x[i][j+1] and x[i][j-1] are the adjacent cells.

Input : mat[][] = {{1, 3},{2, 5}}

Output :2

```
*/
```

```
#include<iostream>
```

```
using namespace std;
```

```
#define loopOuter(i,row) for(int i=0;i<row;i++)
```

```
#define loopInner(j,col) for(int j=0;j<col;j++)
```

```
#define primeLoop(i,sum) for (int i = 2; i <= sum / 2; i++)
```

```
int main ()
```

```
{
```

```
    int count=0,row,col;
```

```
    cout << "Enter number of rows and cloumns" << endl;
```

```
    cin >> row >> col;
```

```
    //Allocating 2D Matrix
```

```
    int **arr = new int*[row];
```

```
    loopOuter(i,row) arr[i] = new int[col];
```

```
    //Input matrix elements
```

```
    cout << "Enter elements " << row*col << " of array:" << endl;
```

```
    loopOuter(i,row) loopInner(j,col) cin >> arr[i][j];
```

```
    loopOuter(i,row){
```

```
        loopInner(j,col)
```

```
    {
```

```
        int sum = 0;
```

```
        bool isPrime = true;
```

```
/*
Search in a row wise and column wise sorted matrix.
Given an n x n matrix
and a number x, find the position of x in the matrix if it is
present in it.
Otherwise, print "Not Found". In the given matrix, every
row and column is
sorted in increasing order. The designed algorithm
should have linear time
complexity

```

```
Input : mat[][] = { {10, 20, 30, 40},{ 15, 25, 35, 45},{27,
29, 37, 48},{32, 33, 39, 50}};
```

```
Element: 29
```

```
Output: found at 2,1
```

```
*/
```

```
#include<iostream>
using namespace std;
#define loopOuter(i,row) for(int i=0;i<row;i++)
#define loopInner(j,col) for(int j=0;j<col;j++)
int main ()
{
    int row, col, ele;
    cout << "Enter number of rows and columns" << endl;
    cin >> row >> col;

    //Allocating 2D Matrix
    int **arr = new int*[row];
    loopOuter(i,row) arr[i] = new int[col];

    //Input matrix elements
    cout << "Enter elements "<< row*col <<" of array:" <<
endl;
    loopOuter(i,row) loopInner(j,col) cin >> arr[i][j];

    cout << "Enter element to be searched" << endl;
    cin >> ele;

```

```
int i = 0; int j = 0;
```

```

endl,
loopOuter(i,row) loopInner(j,col) cin >> arr[i][j];

loopOuter(i,row){
    loopInner(j,col)
{
    int sum = 0;
    bool isPrime = true;

    // i-1, j
    if (i - 1 >= 0)
        sum += arr[i - 1][j];

    // i+1, j
    if (i + 1 < row)
        sum += arr[i + 1][j];

    // i, j-1
    if (j - 1 >= 0)
        sum += arr[i][j - 1];

    // i, j+1
    if (j + 1 < col)
        sum += arr[i][j + 1];

    primeLoop(i,sum){
        if (sum % i == 0) {
            isPrime = false;
            break;
        }
    }
    if (isPrime)
        count++;
    }
}
cout <<"Output: "<< count <<endl;
}

```

```

#include<iostream>
using namespace std;
#define loopOuter(i,row) for(int i=0;i<row;i++)
#define loopInner(j,col) for(int j=0;j<col;j++)
int main ()
{
    int row, col, ele;
    cout << "Enter number of rows and columns" << endl;
    cin >> row >> col;

    //Allocating 2D Matrix
    int **arr = new int*[row];
    loopOuter(i,row) arr[i] = new int[col];

    //Input matrix elements
    cout << "Enter elements "<< row*col <<" of array:" <<
endl;
    loopOuter(i,row) loopInner(j,col) cin >> arr[i][j];

    cout << "Enter elemnt to be searched" << endl;
    cin >> ele;

    int i = 0, j = col-1;
    bool check = true;

    while(i<row && j>=0){
        if(arr[i][j] == ele){
            cout << "Element found at: " << i <<" , "<< j <<
endl;
            check = false;
            break;
        }
        else if(arr[i][j] < ele) i++;
        else j--;
    }
    if(check) cout << "Element not found " <<endl;
}

```

sample input: {  
{1, 1, 0, 0, 1, 1},  
{0, 1, 0, 1, 0, 0},  
{1, 1, 1, 0, 1, 0}}

sample output: 4  
\*/

```
#include<iostream>
using namespace std;
#define loopOuter(i,row) for(int i=0;i<row;i++)
#define loopInner(j,col) for(int j=0;j<col;j++)
int main ()
{
    int count=0,row,col;
    cout << "Enter number of rows and cloumns" << endl;
    cin >> row >> col;

    //Allocating 2D Matrix
    int **arr = new int*[row];
    loopOuter(i,row) arr[i] = new int[col];

    //Input matrix elements
    cout << "Enter elements "<< row*col <<" of array:" <<
endl;
    loopOuter(i,row) loopInner(j,col) cin >> arr[i][j];

    //Logic
    for(int i=0;i<col;i++){

        int sum = 0;

        for(int j=0;j<row;j++) sum+= arr[j][i];

        if (sum % 2 == 1) count++;
    }

    //output
    cout <<"Output: "<< count <<endl;
```