The Prime Cells

Max. Marks: 100

You are given a grid of size $n \times n$ filled with numbers in each of its cells. Now you need to count total cells in the grid such that the sum of the numbers on its top , left , right and bottom cells is a prime number. In case there is no cell in a particular direction assume the number to be as 0.

Input

First line contains a number n as input denoting size of the grid. Next n lines contain n numbers each denoting value of the elements of the grid in each row.

Output

In the output you have to give the count of total cells as described above.

Constraints

$2 \leq n \leq 100$

$1 \leq g[i][j] \leq 100$ where $g[i][j]$ is the value in the grid at i^{th} row and j^{th} column.

SAMPLE INPUT

```
2
1 2
3 4
```

SAMPLE OUTPUT

```
4
```

Explanation

In the given grid if we pick the first element that is **1** then to its right and bottom are 2 and 3 whose sum is 5 and so its prime. Note that to the left there is no element so we consider it as 0 and same goes for the up direction. Checking this for all yields that all the four cells contribute to the count of cells whose sum of the adjacent cell values is prime.

Time Limit:	2.0 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB

Marking Scheme:	Marks are awarded if any testcase passes.
Allowed Languages:	C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic, Kotlin

CODE EDITOR

Enter your code or Upload your code as file. Save C (gcc 5.4.0)

```c
#include<stdio.h>
#define true 1
#define false 0
int  main(){
    int n,m,i,j,sum=0;
    scanf("%d",&n);
    int a[n][n];
    for(i=0;i<n;i++){
        for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);
    }
    m=n-1;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            if((i==0)&&(j==0)){
                if(is_prime(a[0][1]+a[1][0]))
                sum+=1;
                //printf("\n%d\n",a[0][1]+a[1][0]);
            }
            else if((i==0)&&(j==m)){
                if(is_prime(a[0][m-1]+a[1][m]))
                sum+=1;
                //printf("\n%d\n",a[0][m-1]+a[1][m]);
            }
            else if((i==m)&&(j==0)){
                if(is_prime(a[m][1]+a[m-1][0]))
                sum+=1;
                //printf("\n%d\n",a[m][1]+a[m-1][0]);
            }
            else if((i==m)&&(j==m)){
                if(is_prime(a[m][m-1]+a[m-1][m]))
                sum+=1;
                //printf("\n%d\n",a[m][m-1]+a[m-1][m]);
            }
            else if((i==0)||(j==0)){
                if((i==0)&&(j!=0)&&(j!=m)){
                    if(is_prime(a[0][j-1]+a[0][j+1]+a[1][j]))
                    sum+=1;
                    //printf("\n%d\n",a[0][j-1]+a[0][j+1]+a[1][j]);
                }
```

1:1

Provide custom input

Press Ctrl-space for autocomplete suggestions.

COMPILE & TEST SUBMIT

Tip: You can submit any number of times you want. Your best submission is considered for computing total score.