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 Python (python 2.7.6)

```
# A function that does primality test
def is_prime(n):
    if n <= 1:
        return False

    if n <= 3:
        return True

    if n % 2 == 0 or n % 3 == 0:
        return False

    i = 5
    while i * i <= n:
        if (n % i) == 0 or n % (i + 2) == 0:
            return False
        i = i + 6

    return True

def get_the_prime_cells():
    prime_cells = 0
    n = int(raw_input())
    l = []

    i = 0
    while i < n:
        l2 = [int(inp) for inp in raw_input().split()]
        l.append(l2)
        i = i + 1

    i = 0
    while i < n:
        j = 0
        while j < n:
            total = 0

```

```
37              if not i - 1 < 0:
38                  total += l[i - 1][j]
39
```

18:13

Provide custom input

Press Ctrl-space for autocomplete suggestions.

COMPILE & TEST SUBMIT

Submission ID: 10285529 / 3 minutes ago

RESULT: Accepted

Score	Time (sec)	Memory (KiB)	Language
100.0	1.03765	64	Python

Input	Result	Time (sec)	Memory (KiB)	Score
Input #1	✓	0.103143	64	10
Input #2	✓	0.103985	64	10
Input #3	✓	0.102481	64	10
Input #4	✓	0.104178	64	10
Input #5	✓	0.107786	64	10
Input #6	✓	0.103205	64	10
Input #7	✓	0.102143	64	10
Input #8	✓	0.103961	64	10
Input #9	✓	0.102602	64	10
Input #10	✓	0.104167	64	10

Compilation Log

No compilation log for this submission.

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

Your Rating: ★★★★★

COMMENTS (47) SORT BY: Relevance

Join Discussion...

Cancel Post

Brij Desai Edited 3 days ago

Extremely bad problem description !!!

I did it for every possible squares in the grid and counted all the numbers of rectangles in top,left,bottom,right !!

Whereas the actual question asks for primality check of sum of adjacent cells (max 4 such cells) for every cells in the grid.

PS : Problem Description has been edited :)

16 votes ● Reply ● Message ● Permalink

Kashyap Thacker 3 days ago

I could solve it only after reading this comment. Bad description! just consider the adjacent cells! not the whole row or column!!

4 votes ● Reply ● Message ● Permalink

Yastika 3 days ago

thanks

0 votes ● Reply ● Message ● Permalink

Shailesh Suryawanshi a day ago

Thank you dude. I was doing the same. counting each and every damn element. Extremely bad problem description.

0 votes ● Reply ● Message ● Permalink

Khush Gandhi 3 days ago

we need to include all the elements from four sides or just adjacents??

4 votes ● Reply ● Message ● Permalink

Kunal Khatri 3 days ago

only adjacent.

2 votes ● Reply ● Message ● Permalink

Nisarg Bakshi 3 days ago

copy the code

2 votes ● Reply ● Message ● Permalink

Abhishek Inamdar 3 days ago

For sample input:
2
100 100
100 100
My ans in code blocks is 0.
But here it is showing 4.
How is this possible?

2 votes ● Reply ● Message ● Permalink

Rishabh Gupta 21 hours ago

ur right i also get same result for this input,,

0 votes ● Reply ● Message ● Permalink

Sachin Adkar Edited 17 hours ago

i get the correct output, when i run on my turbo c, for all possible inputs. But when i submit here, they that's a wrong answer..

1 vote ● Reply ● Message ● Permalink