Your Find Digits submission got 25.00 points. Share Tweet

Try the Next Challenge | Try a Random Challenge

Find Digits

by HackerRank

Problem Submissions Leaderboard Discussions Editorial Topics

Given an integer, N, traverse its digits (d_1,d_2,...,d_n) and determine how many digits evenly divide N (i.e.: count the number of times N divided by each digit d_i has a remainder of 0). Print the number of evenly divisible digits.

Note: Each digit is considered to be unique, so each occurrence of the same evenly divisible digit should be counted (i.e.: for $N = 111$, the answer is 3).

Input Format

The first line is an integer, T, indicating the number of test cases.
The T subsequent lines each contain an integer, N.

Constraints

$1 \leq T \leq 15$
$0 < N < 10^9$

Output Format

For every test case, count and print (on a new line) the number of digits in N that are able to evenly divide N.

Sample Input

```
2
12
1012
```

Sample Output

```
2
3
```

Explanation

The number 12 is broken into two digits, 1 and 2. When 12 is divided by either of those digits, the calculation's remainder is 0; thus, the number of evenly-divisible digits in 12 is 2.

The number 1012 is broken into four digits, 1, 0, 1, and 2. 1012 is evenly divisible by its digits 1, 1, and 2, but it is *not* divisible by 0 as **division by zero is undefined**; thus, our count of evenly divisible digits is 3.

f y in

Submissions: 103214
Max Score: 25
Difficulty: Easy

Rate This Challenge:

Need Help?

Div Mod

Integer to Array

More

Need Help? Get advice from the discussion forum for this challenge. Or check out the environments page

Current Buffer (saved locally, editable)

Go

```go
package main

import (
    "fmt"
)

func main() {
    testcases := 0
    fmt.Scanf("%d", &testcases)

    for i := 0; i < testcases; i++ {
        counter := 0
        numline := 0
        fmt.Scanf("%d", &numline)
        numslice := digits(numline)
        for i, _ := range numslice {

            if int(numslice[i]) == 0 {
                continue

            } else {
                if numline%int(numslice[i]) == 0 {
                    counter++
                }
            }
        }
        fmt.Println(counter)
    }
}

func digits(dig int) []byte {

    digs := make([]byte, 0, 12)

    for {
        rest := dig / 10
        lastDig := dig - 10*rest
        digs = append(digs, byte(lastDig))
        if dig = rest; dig == 0 {
            break
        }
    }

    // reverse digits
    for i, j := 0, len(digs)-1; i < j; i, j = i+1, j-1 {
        digs[i], digs[j] = digs[j], digs[i]
    }

    return digs

}

```

Line: 1 Col: 13

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

✔ Test Case #0 ✔ Test Case #1

Next Challenge