

# Find Digits



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

 $\begin{array}{l} 1 \leq T \leq 15 \\ 0 < N < 10^9 \end{array}$ 

### 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

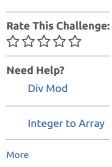
#### **Explanation**

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 ৺ in

Submissions: 103214
Max Score: 25
Difficulty: Easy



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

```
Current Buffer (saved locally, editable) & 🗘
                                                                                   Go
   package main
 2
 3
   import (
 4
        "fmt"
 5
 6
 7 v func main() {
 8
        testcases := 0
 9
        fmt.Scanf("%d", &testcases)
10
11 v
        for i := 0; i < testcases; i++ {
            counter := 0
12
13
            numline := 0
            fmt.Scanf("%d", &numline)
14
15
            numslice := digits(numline)
            for i, _ := range numslice {
16 ▼
17
18 ▼
                if int(numslice[i]) == 0 {
19
                     continue
20
21 ▼
                 } else {
                     if numline%int(numslice[i]) == 0 {
22 ▼
23
                         counter++
24
25
                }
26
27
            fmt.Println(counter)
28
        }
29
30
31 v func digits(dig int) []byte {
32
33
        digs := make([]byte, 0, 12)
34
35 ▼
            rest := dig / 10
36
37
            lastDig := dig - 10*rest
38
            digs = append(digs, byte(lastDig))
39 ▼
            if dig = rest; dig == 0 {
40
                 break
41
42
        }
43
        // reverse digits
44
45 ▼
        for i, j := 0, len(digs)-1; i < j; i, j = i+1, j-1 {
46 ▼
            digs[i], digs[j] = digs[j], digs[i]
47
48
49
        return digs
50
51
   }
52
                                                                                                        Line: 1 Col: 13
```

<u> </u>		Run Code	Submit Code
Congrats, you solved this challenge!			
✔ Test Case #0	✓ Test Case #1		
		Next Challenge	
	Copyright © 2017 HackerRank. All Rights Reserved		

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature