Problem: Find Digits

Given an integer, , traverse its digits (*1*,*2*,...,*n*) and determine how many digits evenly divide  (i.e.: count the number of times  divided by each digit *i* has a remainder of ). 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 , the answer is ).

**Input Format**

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

**Constraints**   
 

**Output Format**

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

**Sample Input**

2

12

1012

**Sample Output**

2

3

**Explanation**

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

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

Solution

void check(long num)

{

long number=num, counter=0;

while(num>0)

{

if(num%10!=0)

{

(number%(num%10)==0 ? counter+=1 : counter+=0);

}

num=num/10;

}

cout<<counter<<endl;

}

int main()

{

int cases;

long num;

cin>>cases;

for(int i=0; i<cases; i++)

{

cin>>num;

check(num);

}

return 0;

}

* Anshul Aggarwal