ECOO '13 R1 P2 - The Luhn Algorithm

Time limit: 30.0s **Memory limit:** 256M

In the 1950's, Hans Peter Luhn invented a method for checking the validity of ID numbers. This method (known as the Luhn Algorithm or the Luhn Formula) is still used today for a number of different purposes, including all major credit card numbers and Social Insurance Numbers.

Here's how the Luhn Algorithm works when checking for a valid ID number:

1. Starting from the right, double every second digit, add up the digits of the result, and total up all the resulting numbers.
2. Add to this total the sum of all the remaining digits.
3. If the result is divisible by 10, the ID number is valid.

Example 1: Validate 42395

Step 1

> $9 \times 2 = 18 \rightarrow 1 + 8 = 9$
>
> $2 \times 2 = 4$
>
> $4 + 9 = 13$

Step 2

> $13 + 4 + 3 + 5 = 25$

Step 3

> 25 is not divisible by 10.

Not valid.

Example 2: Validate 35436

Step 1

> $3 \times 2 = 6$
>
> $5 \times 2 = 10 \rightarrow 1 + 0 = 1$
>
> $1 + 6 = 7$

Step 2

> $7 + 3 + 4 + 6 = 20$

Step 3

> 20 is divisible by 10.

Valid.

Explanation

The last digit of every ID number is the "check digit" and the rest of it is the base number. So in the first example above, `4239` is the base number and `5` is the check digit. When generating ID numbers, you first generate the base number without the final digit, then you figure out what the check digit has to be to make the whole ID number valid.

Specification

The input will contain 5 test cases. Each test case consists of a batch of 5 base numbers (1 to 100 digits each) on one line, each separated by a single space character. Your job is to compute the check digit for each base number in the batch and then output the result as a single 5-digit number.

Sample Input

```
389796 4565280784 8451692334 46 465949539
97699 7392253 54011409 8073542288 303142477
334 349839 12593962 02497993 9468
53173 2901524 2493367526 39094 83530
08080532 5023002 57849 9853641952 027179
```

Sample Output

```
48336
36757
31920
15686
88201
```