Luhn's Checksum Algorithm

Problem ID: luhnchecksu
CPU Time limit: 1 secon
Memory limit: 1024 MB
Difficulty: 1.6

Author: Liam Keliher
Source: 2018 Atlantic Ca Preliminary Contest

In 1954, Hans Peter Luhn, a researcher at IBM, filed a patent describing a simple checksum algorithm for numbers written as strings of base-10 digits. If a number is chosen according to Luhn's technique, the algorithm provides a basic integrity check. This means that with reasonably high probability it can detect whether one or more digits have been accidentally modified. (On the other hand, it provides essentially no protection against intentional modifications.) Most credit card and bank card numbers can be validated using Luhn's checksum algorithm, as can the national identification numbers of several countries (including Canada).

Image by simpson33 (iStock), Used under license

Given a number $n = d_k d_{k-1} \ldots d_2 d_1$, where each d_i is a base-10 digit, here is how to apply Luhn's checksum test:

1. Starting at the *right* end of n, transform every *second* digit d_i (i.e., $d_2, d_4, d_6, \ldots$) as follows:
    1. multiply d_i by 2
    2. if $2 \cdot d_i$ consists of more than one digit, i.e., is greater than 9, add these digits together; this will always produce a single-digit number
2. Add up all the digits of n after the transformation step. If the resulting sum is divisible by 10, *n passes* the Luhn checksum test. Otherwise, *n fails* the Luhn checksum test.

For example, consider the number $n = 1234567890123411$ from Sample Input 1. The first row of Figure 1 gives the original digits of n, and the second row contains the digits of n after the transformation step, with transformed digits shown in bold. The sum of the digits in the second row is

$$2 + 2 + 6 + 4 + 1 + 6 + 5 + 8 + 9 + 0 + 2 + 2 + 6 + 4 + 2 + 1 = 60$$

and since 60 is divisible by 10, n passes the Luhn checksum test.

1	2	3	4	5	6	7	8	9	0	1	2	3	4	1	1
2	2	**6**	4	**1**	6	**5**	8	**9**	0	**2**	2	**6**	4	**2**	1

Figure 1: Application of Luhn's algorithm to $n = 1234567890123411$

Input

The first line of input contains a single integer T ($1 \leq T \leq 100$), the number of test cases. Each of the following T lines contains a single test case consisting of a number given as a string of base-10 digits (`0`–`9`). The length of each string is between 2 and 50, inclusive, and numbers may have leading (leftmost) zeros.

Output

For each test case, output a single line containing "`PASS`" if the number passes the Luhn checksum test, or "`FAIL`" if the number fails the Luhn checksum test.

Sample Input 1

```
3
00554
999
1234567890123411
```

Sample Output 1

```
PASS
FAIL
PASS
```