Luhn Algorithm

The input consists of a bunch of lines of sample credit card numbers, each followed by a newline. Your task is to write a program that validates each credit card number according to the Luhn algorithm, which will be explained below. You should output the same number followed by a comma, and the string "TRUE" if the number is valid or "FALSE" if the number is not valid.

The formula for validating a credit card, the Luhn algorithm, can be implemented as follows:

1. Sum the digits in odd-numbered positions (counting from the right-most side of the number.)
2. Double each digit in an even-numbered position (again, counting from the right-most side of the number), sum the digits of the resulting values (note: not the values themselves.)
3. Add the sums from steps one and two.
4. If that total is evenly divisible by 10 (no remainder) the card number is considered valid.

Note that valid cards must also have a valid prefix taken from this list: [51, 52, 53, 54, 55, 4].

We are also assuming that all cards numbers in the input file are a length of 16.

Example 1

Given an example (valid) card number of 4652360088404638.

1. Odd digits: 8 + 6 + 0 + 8 + 0 + 6 + 2 + 6 = 36
2. Even digits 3 4 4 8 0 3 5 4
   Doubled 6 8 8 16 0 6 10 8
   Sum of digits 6 + 8 + 8 + 1 + 6 + 0 + 6 + 1 + 0 + 8 = 44
3. Total of sums from steps 1 and 2: 36 + 44 = 80
4. 80 % 10 = 0

The remainder at the end was zero, so the number is considered valid.

Example 2

Given an example (invalid) card number of 5370182444652350.

1. Odd digits: 0 + 3 + 5 + 4 + 4 + 8 + 0 + 3 = 27
2. Even digits 5 2 6 4 2 1 7 5
   Doubled 10 4 12 8 4 2 14 10
   Sum of digits 1 + 0 + 4 + 1 + 2 + 8 + 4 + 2 + 1 + 4 + 1 + 0 = 28
3. Total of sums from steps 1 and 2: 27 + 28 = 55
4. 55 % 10 = 5

The remainder at the end was 5, so the number is not considered valid.

Note: The ¶ symbol in the examples below represents a new-line character.

Sample Input

```
4652360088404638¶
```

```
5370182444652350¶
```

Sample Output

```
4652360088404638,TRUE¶
5370182444652350,FALSE¶
```