Problem B: Credit Card Parity Checker

The Luhn algorithm, also known as the *modulus 10* or *mod 10* algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers. The formula verifies a number against its included check digit, which is usually appended to a partial account number to generate the full account number. This account number must pass the following test:

1. Starting from the rightmost digit (which is the check digit) and moving left, double the value of every second digit. For any digits that then become 10 or more, take the two numbers and add them together. For example, 1111 becomes 2121, while 8763 becomes 7733 (from $2 \times 6 = 12 \rightarrow 1 + 2 = 3$ and $2 \times 8 = 16 \rightarrow 1 + 6 = 7$).
2. Add all these digits together. For example, if 1111 becomes 2121, then $2+1+2+1 = 6$; and 8763 becomes 7733, so $7 + 7 + 3 + 3 = 20$.
3. If the total ends in 0 (put another way, if the total modulus 10 is congruent to 0), then the number is valid according to the Luhn formula; otherwise it is not valid. So, 1111 is not valid (as shown above, it comes out to 6), while 8763 is valid (as shown above, it comes out to 20).

Input

Input will consist of a list of numbers, one per line, where each number is composed of one or more digits. Your program should read one number at a time and determine if the number is correct according to the Luhn algorithm. It will continue reading these numbers until it reads a 0. Your program should not produce any output for this 0.

Output

For each correct number based on the above description read by your program, print `Number is correct` and for each wrong number, output `Number is not correct`.

Sample input (available as file "B.in"):

```
1626846262414
638796128
819771997
513545716246
96557425276437
346692
0
```

Sample output (available as file "B.out"):

```
Number is not correct
Number is correct
Number is not correct
Number is not correct
Number is correct
Number is correct
```