PROBLEM 2 - CHANGE

A small local shop is having a problem because the assistants find it hard to work out how much change to give to customers. You have been asked to help them by writing a program that does all the work!

Notes and coins available are as follows:

Notes: $20, $10, $5, $2, $1.

Coins: 50c, 20c, 10c, 5c.

As the smallest coin available is 5c, the cost of the purchase may need to be rounded to the nearest 5c, using the so-called Swedish rounding method. The rules for rounding are as follows:

1 or 2 cents – rounded down to 0.

3 or 4 cents – rounded up to 5.

6 or 7 cents – rounded down to 5.

8 or 9 cents – rounded up to 10.

The program must work out the change required and specify the notes and coins to use. In each case, the smallest possible number of notes and coins must be used.

INPUT FORMAT

Each line of input will represent a single transaction, and will contain 2 decimal numbers in the range 0.05 to 1000.00, each with two digits after the decimal point, and separated by a single space. The first number is the cost of a purchase, the second the amount the customer offers at the till. As mentioned, the cost of the purchase may need to be rounded, and, of course, the amount offered by the customer is a multiple of 5 cents. A line consisting of two 0.00 numbers marks the end of the input.

SAMPLE INPUT

```
20.03 20.00
20.07 20.05
20.08 25.00
0.09 0.10
0.00 0.00
```

OUTPUT FORMAT

Your program must output one line for each transaction. Where the amount of money offered by the customer is not enough to cover the rounded purchase price, your program must output

```
Not enough money offered.
```

Where the amount of money offered by the customer is exactly the rounded purchase price, your program must output

```
Exact amount.
```

In all other cases output the sequence describing the change. Each sequence item starts with a note or coin value in the format described earlier (e.g., $2 or 10c), followed by a multiplication sign (i.e., an asterisk, '*') and ends with a repetition count (a number ≥ 1). Items are listed in order of decreasing values and are separated by single spaces.

SAMPLE OUTPUT

```
Not enough money offered.
Exact amount.
$2*2 50c*1 20c*2
Exact amount.
```