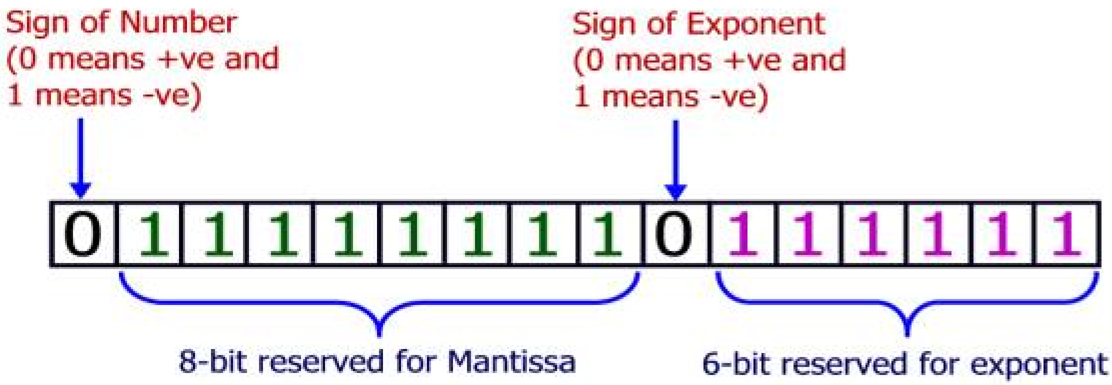
Floating-point numbers are represented differently in computers than integers. That is why a 32-bit floating-point number can represent values in the magnitude of 10^{38} while a 32-bit integer can only represent values as high as 2^{32}.

Although there are variations in the ways floating-point numbers are stored in Computers, in this problem we will assume that floating-point numbers are stored in the following way:



Floating-point numbers have two parts mantissa and exponent. M-bits are allotted for mantissa and E bits are allotted for exponent. There is also one bit that denotes the sign of number (If this bit is 0 then the number is positive and if it is 1 then the number is negative) and another bit that denotes the sign of exponent (If this bit is 0 then exponent is positive otherwise negative). The value of mantissa and exponent together make the value of the floating-point number. If the value of mantissa is m then it maintains the constraints $\frac{1}{2} \le m < 1$. The left most digit of mantissa must always be 1 to maintain the constraint $\frac{1}{2} \le m < 1$. So this bit is not stored as it is always 1. So the bits in mantissa actually denote the digits at the right side of decimal point of a binary number (Excluding the digit just to the right of decimal point)

In the figure above we can see a floating-point number where $M = 8$ and $E = 6$. The largest value this floating-point number can represent is (in binary) $0.111111111_2 \times 2^{111111_2}$. The decimal equivalent to this number is: $0.998046875 \times 2^{63} = 9205357638345293824_{10}$. Given the maximum possible value represented by a certain floating point type, you will have to find how many bits are allotted for mantissa (M) and how many bits are allotted for exponent (E) in that certain type.

Input

The input file contains around 300 line of input. Each line contains a floating-point number F that denotes the maximum value that can be represented by a certain floating-point type. The floating point number is expressed in decimal exponent format. So a number AeB actually denotes the value $A \times 10^B$. A line containing '`0e0`' terminates input. The value of A will satisfy the constraint $0 < A < 10$ and will have exactly 15 digits after the decimal point.

Output

For each line of input produce one line of output. This line contains the value of M and E. You can assume that each of the inputs (except the last one) has a possible and unique solution. You can also assume that inputs will be such that the value of M and E will follow the constraints: $9 \ge M \ge 0$ and $30 \ge E \ge 1$. Also there is no need to assume that $(M + E + 2)$ will be a multiple of 8.

Sample Input

```
5.699141892149156e76
9.205357638345294e18
0e0
```

Sample Output

```
5 8
8 6
```