

G Expectant Dice (dice.{c,cc,java})

G.1 Description

Games involving large assortments of dice have developed their own notation for describing common die rolls. At the basic level, we have notation like “d6,” which means “roll a 6-sided die.” A common extension to this is something like “5d8,” which means “roll 5 8-sided dice and add up the results.”

What if we were to extend the notation even further? Let’s say that “(d6)(d8)” means “roll a 6-sided die, and use the result as the number of 8-sided dice to roll and add together.” Let’s also say that “d(d4)” means “roll a 4-sided die, and use the result as the number of sides on the die to roll for the final result.” In general, we can say that “dX” means “roll an X-sided die,” while “YdX” means “roll Y X-sided dice and add up the results.” We will say that all valid strings in our notation have the following properties:

- The only characters in the string are the letter d and the digits 1 through 9.
- The string cannot end with a d.
- No two digits are adjacent in the string; they are always separated by at least one d.
- The string is of length at least 2.

We can see that some strings can have ambiguous interpretations. For example, dd5d9 can mean d((d5)(d9)), (d(d5))(d9), or d(d(5d9)). All of these expressions will give different results; moreover, the expected value of each disambiguated expression is different. In fact, the expected values of these three disambiguations are 8, 10, and 7, respectively. Given a potentially ambiguous die string, can you find the maximum and minimum expected values over all possible disambiguations of the string?

G.2 Input

The input consists of multiple test cases. Each test case consists of a single line containing a valid die string, D , whose length is between 2 and 20 inclusive. The input is followed by a case with $D = 0$, which should not be processed. For example:

```
d6
dd5d9
0
```

G.3 Output

For each test case, print a single line containing two numbers min and max , separated by a space, where min is the minimum possible expected value of the die string, and max is the maximum possible expected value of the die string. Each number should be rounded to 4 decimal places. For example:

```
3.5000 3.5000
7.0000 10.0000
```