553 Simply proportion

Wou will probably have made a document on a word processor. When the job was done and you wanted to print that document you sometimes have the option to print the document with proportional spacing. This means that gaps between letters and words are filled with very small spaces (as small as your printer can handle) in such a way that you do not see a ragged appearance of a line but instead the line is left and right justified on your printout. This could also be done by filling each line with normal spaces, but this usually produces big gaps between words. In the following we call these small spaces **dots**.

In the exercise you are asked to produce an algorithm that fills lines with dots in such a manner that the line has a certain length (measured in dots). We will provide you with lines which contain spaces and certain letters. Each character has a width that can be measured in dots. In order to keep this exercise simple we will only use a subset of letters. These letters are the following:

Character	**Width**
A	18
B	17
I	10
M	20
S	16
Y	13
'space'	Variable

The minimum number of dots between letters in a word is 3. The number of dots that separates letters in a word is given by the greatest **possible** number that is equal to or smaller than 1/3 of the minimum number of dots that separate words (rounded down to the nearest integer). The minimum number of dots for a single space is 10(there is no upper limit for the number of dots in a space).

Note: the begin and end of a line may not contain (empty) dots.

It can (and according to Murphy's law it will) happen that you are left with a number of dots that can't be equally divided in the gaps between letters or words. These leftovers must be equally divided over the spaces between the words beginning from the end of the line.

Input

The input for your program is a text-file. Each input consists of two lines. On the first line there is an integer (say N) that tells you the desired length of the line (measured in dots) that will be your output. This integer has a maximum value of 5000. The second line consists of the input that you have to reformat to the desired length (each word is separated from the other by one space). Each line has at least two words on it and the maximum number of characters on this line is 80. There are no spaces after the last word. It is given that it is possible to fill the given line with dots so that the resulting line has a length of N dots. The last two lines of the input are given by:

```
0
SYMIBA
```

Your program must not perform any action on this last input, it is simply there to mark the end of the input.

Output

Your program must convert the given line to the desired length (N dots). The output is a text-file. Each line corresponds to the second line of each input pair with the number of dots between the letters and words. The spaces between letters and words are denoted in a special manner. If there are 3 dots between the letters `A` and `B` this is denoted by `A/(3)B` in your output. Your output may not contain any spaces, all these spaces must be converted to dots.

Sample Input

```
250
AIM SSY ABABA
200
SSSS AAAA
130
AA B AA
0
SYMBIA
```

Sample Output

```
A/(4)I/(4)M/(18>S/(4)S/(4)Y/(19)A/(4)B/(4)A/(4)B/(4)A
S/(7)S/(7)S/(7)S/(22)A/(7)A/(7)A/(7)A
A/(5)A/(16)B/(16)A/(5)A
```