# B. Text Document Analysis

1 second, 256 megabytes

Modern text editors usually show some information regarding the document being edited. For example, the number of words, the number of pages, or the number of characters.

In this problem you should implement the similar functionality.

You are given a string which only consists of:

- uppercase and lowercase English letters,
- underscore symbols (they are used as separators),
- parentheses (both opening and closing).

It is guaranteed that each opening parenthesis has a succeeding closing parenthesis. Similarly, each closing parentheses has a preceding opening parentheses matching it. For each pair of matching parentheses there are no other parenthesis between them. In other words, each parenthesis in the string belongs to a matching "opening-closing" pair, and such pairs can't be nested.

For example, the following string is valid: "`_Hello_Vasya(and_Petya)__bye_(and_OK)`".

*Word* is a maximal sequence of consecutive letters, i.e. such sequence that the first character to the left and the first character to the right of it is an underscore, a parenthesis, or it just does not exist. For example, the string above consists of seven words: "`Hello`", "`Vasya`", "`and`", "`Petya`", "`bye`", "`and`" and "`OK`". Write a program that finds:

- the length of the longest word outside the parentheses (print `0`, if there is no word outside the parentheses),
- the number of words inside the parentheses (print `0`, if there is no word inside the parentheses).

## Input

The first line of the input contains a single integer $n$ ($1 \le n \le 255$) — the length of the given string. The second line contains the string consisting of only lowercase and uppercase English letters, parentheses and underscore symbols.

## Output

Print two space-separated integers:

- the length of the longest word outside the parentheses (print `0`, if there is no word outside the parentheses),
- the number of words inside the parentheses (print `0`, if there is no word inside the parentheses).

| input |
| --- |
| 37<br>_Hello_Vasya(and_Petya)__bye_(and_OK) |
| **output** |
| 5 4 |

| input |
| --- |
| 37<br>_a_(_b___c)__de_f(g_)__h__i(j_k_l)m__ |
| **output** |
| 2 6 |

**input**

```
27
(LoooonG)__shOrt__(LoooonG)
```

**output**

```
5 2
```

**input**

```
5
(___)
```

**output**

```
0 0
```

In the first sample, the words "`Hello`", "`Vasya`" and "`bye`" are outside any of the parentheses, and the words "`and`", "`Petya`", "`and`" and "`OK`" are inside. Note, that the word "`and`" is given twice and you should count it twice in the answer.