Number of c's with an a before and without b between, and number of d's with an a after and without b between X17557_en

The input of this exercise is a sequence of characters over `{a,b,c,d}`. We want to count two things:

- We want to count the number of `c`'s satisfying the following: before this `c` there is at least one `a`. Moreover, if we consider the closest `a` to this `c` and before this `c`, then, between both letters there is no `b`. For example, in the following word we have boldfaced the `c`'s that must be counted: `cdbdad`**`cc`**`bdca`**`c`**`bbdcdabcd`.
- We want to count the number of `d`'s satisfying the following: after this `d` there is at least one `a`. Moreover, if we consider the closest `a` to this `d` and after this `d`, then, between both letters there is no `b`. For example, in the following word we have boldfaced the `d`'s that must be counted: `cdb`**`d`**`adccb`**`d`**`cacbbb`**`d`**`c`**`d`**`abcd`.

Input

The input is a sequence of characters over `{a,b,c,d}`, in just one line.

Output

The output has two natural numbers, in one line, and separated by a white space:

- the number of `c`'s that have an `a` before and no `b` between them.
- the number of `d`'s that have an `a` after and no `b` between them.

Sample input 1

```
cdbdadccbdcacbbdcdadacacadbcd
```

Sample output 1

```
5 5
```

Sample input 2

```
bbabddccdcdcddddcdcadccbddcabbbbbaaaccdcdcc36d23adcdcccdcaaddcdcdddccddcbbaadddcacdddcddbacdcccac
```

Sample output 2

Observation

It is not allowed to use any massive storage data structure, not even `string`. Read and treat the input character by character.
Assessment over 10 points:

- Slow solution: 5 points.
- Fast solution: 10 points.

We understand as fast solution one being correct, with linear cost and able to overcome both the public and private tests. We understand as slow solution one not being fast, but correct and able to overcome the public tests.

Problem information

Author: PRO1

Generation: 2026-01-25T14:02:10.395Z