

Search for a polynomial

In a given text, we have to detect all occurrences of a polynomial in one variable x. For introduction, we provide short informal (and by no means complete) characterization of a polynomial. For full description of the polynomial structure see Table 1 below. Informally, a polynomial in variable x is a sequence of terms, terms are separated by addition or subtraction signs. A term consists of a constant followed by a power of x. Sometimes, either the constant or the power of x might not be present in the term. A constant is an integer constant or a float constant. The length of constants and exponents in a polynomial is not limited, the length of a polynomial itself is not limited too. A polynomial does not contain any of the following: Spaces or blank characters, parentheses, multiplication or division signs. It also might not contain any occurrence of variable x. Some examples of polynomials are listed below.

Example A. Strings representing polynomials in format used in this problem.

```
0
1.0-0.0+0.1
-x+1.2
x^4-4x^3+6x^2-4x+1
0.01-x-x-x-0.001E12345x^54321-1.1E1x^1-x^0-0.01
```

Table 1. Regular grammar generating polynomials in this problem. The grammar is expressed in Backus-Naur Form. A polynomial is represented by the start symbol <POLY>.

```
<POLY> ::= <TERMS> | <minus><TERMS>
<TERMS> ::= <TERM> | <TERM><plusmin><TERMS>
<TERM> ::= <CONST> | <VAR_EXP> | <CONST><VAR_EXP>
<CONST> ::= <FLOAT> | <INT>
<VAR_EXP> ::= <var> | <var><pow><EXPON>

<FLOAT> ::= <INT><dot><DIG_SEQ> | <INT><dot><DIG_SEQ><e><EXPON>
<EXPON> ::= <INT> | <plusmin><POS_INT>
<INT> ::= 0 | <POS_INT>
<POS_INT> ::= <dig_1_9> | <dig_1_9><DIG_SEQ>
<DIG_SEQ> ::= <dig_0_9> | <dig_0_9><DIG_SEQ>

<dig_0_9> ::= 0 | <dig_1_9>
<dig_1_9> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<dot> ::= .           // ascii 46, full stop / period / dot character
<e> ::= E
<minus> ::= -          // ascii 45, minus character
<pow> ::= ^            // ascii 94, caret character
<plusmin> ::= + | -    // ascii 43 or 45, plus or minus character
<var> ::= x
```

The task

Count the number of occurrences of polynomials of maximum length in the given text. A polynomial in the text has maximum length if it is not a substring of any longer polynomial in the same text.

Implementation notice

It turns out that correct usage of regular expressions (regex) libraries might be sometimes tricky when the patterns are overlapping in the text. Therefore, your solution should not depend on any regular expressions support of your language.

Input

The input consists of single text line which contains a string without spaces or blank characters. The string contains only printable ascii characters (code 33–126). The maximum length of the string is 10^7 .

Output

The output is one line containing one integer - the number of occurrences of polynomials of maximum length in the given text.

Example 1

Input

```
yy2x^3yy45yy000yy
```

Output

```
5
```

The text in Example 1 contains polynomial $2x^3$, constant polynomial 45, and three constant 0 polynomials. Note that any leading zeros

in any constant violate the definition above.

Example 2

Input

```
###x+2.303.4+x###
```

Output

```
2
```

The text in Example 2 contains two polynomials of maximum length: $x+2.303$ and $303.4+x$. The polynomials overlap in the text. Note that e.g. polynomial $x+2.3$ is not considered because it does not fulfill the maximality condition.

Example 3

Input

```
x+y-x+x^1-20.02E-5xy+3zx-12+z
```

Output

```
4
```

The text in Example 3 contains four polynomials of maximum length: x , $-x+x^1-20.02E-5x$, 3 , $x-12$.

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

[Link to public data set](#)