CamelCase ☆

Editorial by Nabila_ahmed

Each word begins with a capital letter, so you can solve this problem by counting the number of capitalized characters and adding 1 to that number. This works because each capital letter signifies the start of a word, with the exception of the very first word which is lowercase (hence the $+1$).

Problem Setter's code:

C++

```
#include <bits/stdc++.h>
#include<assert.h>

using namespace std;


void solution() {

   string str;
   cin >> str;
   int len = str.size();
   int ans = 1;
   for(int i = 0; i < len; i++){
       if(str[i] >= 'A' && str[i] <= 'Z') {
           ans++;
       }
   }
   cout<<ans<<endl;
}


int main() {

        solution();

    return 0;
}
```

Tested by Shafaet

Problem Tester's code:

Python 2

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
s = raw_input()
ans = 1
assert len(s) >= 1 and len(s) <= 100000
for c in s:
    if ord(c) >= ord('A') and ord(c) <= ord('Z'):
        ans = ans + 1

print ans
```

Java (AllisonP)

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String s = scan.next();
        scan.close();

        // use a regex matching to split the string on capital letters
        // the resulting array contains contiguous sections of lowercase letters
        String[] words = s.split("[A-Z]");
        // this works because the problem states that each word has at least 2 characters, and we know that the first charact

        System.out.println(words.length);
    }
}
```

Feedback

Was this editorial helpful?

Yes No