

题目描述：

给定一个字符串，只包含字母和数字，按要求找出字符串中的最长（连续）子串的长度，字符串本身是其最长的子串，子串要求：

- 1、 只包含 1 个字母($a\sim z$, $A\sim Z$)，其余必须是数字；
- 2、 字母可以在子串中的任意位置；

如果找不到满足要求的子串，如全是字母或全是数字，则返回 -1。

输入描述：

字符串(只包含字母和数字)

输出描述：

子串的长度

示例 1

输入：

abC124ACb

输出：

4

说明：

满足条件的最长子串是 C124 或者 124A，长度都是 4

示例 2

输入:

a5

输出:

2

说明:

字符串自身就是满足条件的子串，长度为 2

示例 3

输入:

aBB9

输出:

2

说明:

满足条件的子串为 *B9*，长度为 2

示例 4

输入:

abcdef

输出:

-1

说明:

没有满足要求的子串，返回 -1

```
import java.util.Scanner;
```

```

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        while (in.hasNextLine()) { // 注意 while 处理多个 case
            String input = in.nextLine();
            char[] src = input.toCharArray();
            int maxLength = -1;
            int left = 0, right = 0;

            boolean hasDigit = false;
            boolean hasChar = false;
            while (right < src.length) {
                if (src[right] >= '0' && src[right] <= '9') {
                    hasDigit = true;
                    maxLength = Math.max(maxLength, right - left + 1);
                } else {
                    hasChar = true;
                    for (int i = right - 1; i >= left; i--) {
                        if (!(src[i] >= '0' && src[i] <= '9')) {
                            left = i + 1;
                            break;
                        }
                    }
                    maxLength = Math.max(maxLength, right - left + 1);
                }

                right++;
            }

            if (!hasDigit || !hasChar) {
                System.out.println(-1);
            } else {
                System.out.println(maxLength);
            }
        }
    }
}

```