

寻找符合要求的最长子串

知识点 [双指针](#)

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

给定一个字符串 s ，找出这样一个子串：

- 1) 该子串中的任意一个字符最多出现2次；
- 2) 该子串不包含指定某个字符；

请你找出满足该条件的最长子串的长度。

输入描述：

第一行为要求不包含的指定字符，为单个字符，取值范围[0-9a-zA-Z]

第二行为字符串 s ，每个字符范围[0-9a-zA-Z]，长度范围[1,10000]

输出描述：

一个整数，满足条件的最长子串的长度；如果不存在满足条件的子串，则返回0

示例1

输入:

D
ABC123

输出:

6

示例2

输入:

D
ABACA123D

输出:

7

解题思路:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        char c = sc.nextLine().charAt(0);  
        String s = sc.nextLine();  
  
        int left = 0;    //子串的起始位置  
        int max = 0;  
        /**  
         * key: 字符  
         * value: 字符出现的次数  
         */  
        Map<Character, Integer> map = new HashMap<>();  
        for(int i=0; i<s.length(); i++){  
            char temp = s.charAt(i);  
            if(temp == c){  
                map.clear();    //指定字符之前的数据需要清空
```

```

        left = i + 1; //指定字符本身不做计数，所以需要+1
        continue;
    }

    map.put( temp, map.getDefault(temp, 0) + 1);
    while (map.get(temp) == 3){        //字符出现第二次之前的字符都需要删除
        char rmStr = s.charAt(left);
        left ++;
        map.put( rmStr, map.get(rmStr) - 1);    //起始位置后移，字符个数也需要
-1
    }

    max = Math.max( max, i - left + 1);
}

System.out.println(max);
}
}

```