

题目描述

输入两个字符串s和t，都只包含英文小写字母。S长度<=100，t长度<=500,100，判定S能否匹配t的有效子串。

判定规则：

S中每个字符必须在t中能匹配到（可以不连续），且S是t中字符串的有序子串与S中每个字符匹配。

（例如，S="ac"，t="abcde"的一个子字符串有效字符串是a c，a，而"ac"不是有效字符串，没有有效字符串是a，e）

输入描述

输入两个字符串s和t，都只包含英文小写字母。S长度<=100，t长度<=500,100，判定S是否，再输入t，每个字符串占一行。

输出描述

S串匹配一个有效字符串在t中的位置。（位置从0开始计算，无有效字符串返回-1）

用例

输入	ac
abcde	
输出	4
说明	无
输入	bd
abcde	
输出	-1
说明	无

题目解析

本题可以利用双指针来解决。

定义两个指针i,j，分别指向S，t字符串的索引0位置。

- 当i和j指向字符串相等时，i++，j++
- 当i和j指向字符串不相等时，j++

当i==s.length()==t.length()时

说明此时，i==s.length，j==t.length，在字符串中找到了所有S字符串，且字符串是一个字符串在t中的位置是0，-1，否则，返回-1。

两例示意图如下：



Java算法源码

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         String s = sc.next();
8         String t = sc.next();
9
10        System.out.println(getResult(s, t));
11    }
12
13    public static int getResult(String s, String t) {
14        int i = 0;
15        int j = 0;
16
17        while (i < s.length() && j < t.length()) {
18            if (s.charAt(i) == t.charAt(j)) {
19                i++;
20                j++;
21            }
22            if (i == s.length()) return j - 1;
23            else return -1;
24        }
25    }
26 }
```

JavaScript源码

```
1 function main(s, t) {
2     let result = getResult(s, t);
3     console.log(result);
4 }
5
6 const rl = require('readline').createInterface({
7     input: process.stdin,
8     output: process.stdout,
9 });
10
11 const lines = [];
12 rl.on('line', (line) => {
13     lines.push(line);
14 });
15
16 if (lines.length > 0) {
17     console.log(getResult(lines[0], lines[1]));
18     lines.length = 0;
19 }
20
21 // 运行结果
22
23 function getResult(s, t) {
24     let i = 0;
25     let j = 0;
26
27     while (i < s.length && j < t.length) {
28         if (s[i] == t[j]) {
29             i++;
30             j++;
31         }
32         if (i == s.length) return j - 1;
33         else return -1;
34     }
35 }
```

Python算法源码

```
1 i = 0
2 j = 0
3 s = input()
4 t = input()
5
6 while i < len(s) && j < len(t):
7     if s[i] == t[j]:
8         i += 1
9         j += 1
10    else:
11        j += 1
12
13    if i == len(s):
14        return j - 1
15    else:
16        return -1
17
18 # 运行结果
19 print(getResult(s, t))
```