

题目描述：

给定字符串 *target* 和 *source*，判断 *target* 是否为 *source* 的子序列。

你可以认为 *target* 和 *source* 中仅包含英文小写字母。字符串 *source* 可能会很长（长度 $\sim 500,000$ ），而 *target* 是个短字符串（长度 ≤ 100 ）。

字符串的一个子序列是原始字符串删除一些（也可以不删除）字符而不改变剩余字符相对位置形成的新字符串。（例如，"abc"是"aebycd"的一个子序列，而"ayb"不是）。

请找出最后一个子序列的起始位置。

输入描述：

第一行为 *target*，短字符串（长度 ≤ 100 ）

第二行为 *source*，长字符串（长度 $\sim 500,000$ ）

输出描述：

最后一个子序列的起始位置，即最后一个子序列首字母的下标

补充说明：

若在 *source* 中找不到 *target*，则输出 -1

示例 1

输入：

```
abc
abcaybec
```

输出：

```
3
```

说明：

这里有两个 *abc* 的子序列满足，取下标较大的，故返回 3

```
#include <bits/stdc++.h>
using namespace std;
const int N = 5e5 + 7;
const int M = 1e2 + 7;
char tag[M], src[N];
int n, m, p;
int main() {
    scanf("%s%s", tag + 1, src + 1);
```

```
n = strlen(tag + 1), m = strlen(src + 1);
p = n;
for(int i = m; i >= 1; i--)
if(tag[p] == src[i])
{
    // cout << p << " " << i << endl;
    p--;
    if(p == 0) {cout << i - 1 ; return 0;}
}
cout << -1;
return 0;
}
// 64 位输出请用 printf("%lld")
```