

## 判断字符串子序列

### 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  $\leq 100$ ）。

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  $\leq 100$ ）

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

```
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);  
  
        String target=in.next();  
  
        int res=-1;  
  
        if(in.hasNext()){  
  
            String source=in.next();  
  
            for(int i=source.length()-1;i>=0;i--){  
  
                int j=0,k=i;  
  
                while(k<source.length()&&j<target.length()){  
  
                    if(source.charAt(k)==target.charAt(j)){  
  
                        j++;  
  
                    }  
  
                    k++;  
  
                }  
  
                if(j==target.length()){  
  
                    res=i;  
  
                    break;  
  
                }  
  
            }  
  
        }  
  
        System.out.print(res);  
    }  
}
```

}

}