

题目描述：

C 语言有一个库函数：`char *strstr(const char *haystack, const char *needle)`，实现在字符串 `haystack` 中查找第一次出现字符串 `needle` 的位置，如果未找到则返回 `null`。

现要求实现一个 `strstr` 的增强函数，可以使用带可选段的字符串来模糊查询，与 `strstr` 一样返回首次查找到的字符串位置。

可选段使用“`[]`”标识，表示该位置是可选段中任意一个字符即可满足匹配条件。比如“`a[bc]`”表示可以匹配“`ab`”或“`ac`”。

注意目标字符串中可选段可能出现多次。

输入描述：

与 `strstr` 函数一样，输入参数是两个字符串指针，分别是源字符串和目标字符串。

输出描述：

与 `strstr` 函数不同，返回的是源字符串中，匹配子字符串相对于源字符串地址的偏移（从 0 开始算），如果没有匹配返回 -1。

补充说明：

源字符串中必定不包含“`[]`”；目标字符串中“`[]`”必定成对出现，且不会出现嵌套。

输入的字符串长度在[1,100]之间。

示例 1

输入：

abcd

b[cd]

输出：

1

说明：

相当于是源字符串中查找 `bc` 或者 `bd`，`bc` 子字符串相对于 `abcd` 的偏移是 1

```
string = input()
```

```
target = input()
```

```
def func(string, target):
```

```
    if "[" not in target:
```

```
        if target in string:
```

```
            print(string.index(target))
```

```
            return
```

```
        print(-1)
```

```
        return
```

```
    target_copy = target
```

```
    count = target.count("[")
```

```
    special = []
```

```
    py = 0
```

```
    while "[" in target:
```

```
        special.append((target.index("[") + py, target.index("]") + py))
```

```
        py = target.index("]") + 1
```

```

        target = target[py:]
#        print(special)

def in_special(index):
    for start, end in special:
        if start < index < end:
            return start, end
    return 0, 0
def generate_target_list(target):
    res = []
    index = 0
    length = len(target)
    while index < length:
        start, end = in_special(index)
        if target[index] not in "[]" and start == 0 and end == 0:
            res = [item + target[index] for item in res]
            index += 1
        elif not (start == 0 and end == 0):
            tmp = []
            for i in range(start+1, end):
                res_i = [item + target[i] for item in res]
                tmp.extend(res_i)
            res = tmp
            index = end
        else:
            index += 1
    return res
target_list = generate_target_list(target_copy)
res = 200
for item in target_list:
    if item in string:
        res = min(res, string.index(item))

print(-1 if res == 200 else res)

```

```

func(string, target)

```