

# 字符串程序的综合分析——子串的指针版简单检索

2022 年 8 月 16 日

1. 阅读程序，回答相应问题。

---

```
1  #include <stdio.h>
2  /*子串检索*/
3  const int MAX_LEN = 1024;
4  int main()
5  {
6      char s1[MAX_LEN];
7      char s2[MAX_LEN];
8      printf("Input a string to be searched:\n");
9      gets(s1);
10     printf("Input a string to be located:\n");
11     gets(s2);
12     int t;
13     char *p1, *p2;
14     for(t = 0; *(s1 + t); t++)
15     {
16         p1 = s1 + t;
17         p2 = s2;
18         while((*p2) && (*p2 == *p1))
19         {
20             p1++;
21             p2++;
22         }
23         if(!(*p2))
24         {
25             printf("The first occurrence of %s is at index %d in %s\n", s2, t, s1);
26             return 0;
27         }
```

```
28     }
29     printf("%s not found\n", s2);
30     return 0;
31 }
```

---

- (a) 试分别就以下两种情况，分析以上程序的运行，并写出输出结果。
- i. 输入 `s1` 为 “The goose is good.”，输入 `s2` 为 “good”；
  - ii. 输入 `s1` 为 “Then she got a sheep.”，输入 `s2` 为 “shed”。
- (b) 假设用户输入 `s1` 为 “The goose is good.”，输入 `s2` 为 “good”。计算第 14 行的 `t++`、第 18 行的 `*p2==*p1` 和第 20 行的 `p1++` 分别被执行或判断的次数。
- (c) 假设用户输入 `s1` 为 “Then she got a sheep.”，输入 `s2` 为 “shed”。计算第 14 行的 `t++`、第 18 行的 `*p2==*p1` 和第 20 行的 `p1++` 分别被执行或判断的次数。