

第 11 次 C 上机

字符串

张晓平

武汉大学数学与统计学院

2017 年 5 月 9 日

例 1 设计并测试一个函数 `my_strchr()`，其功能是搜索由函数的第一个参数指定的字符串，在其中查找由函数的第二个参数指定的字符的第一个出现的位置。如果找到，返回指向这个字符的指针；如果没有找到，返回 `NULL`。在一个使用循环语句为这个函数提供输入的完整程序中进行测试。

```
1 #include<stdio.h>
2 char * my_strchr(const char * str, int c);
3
4 int main(void) {
5     char * str = "0123456789012345678901234567890";
6     char * p;
7     char c = '5';
8
9     p = my_strchr(str, c);
10    if (p == NULL)
11        printf("Cannot find the character %c!\n", c);
12    else
13        printf("Find the character %c\n", *p);
14
15    return 0;
16 }
17
```

```
18 char * my_strchr(const char * str, int c)
19 {
20
21     while(*str && *str != c)
22         str++;
23     if (*str == c)
24         return ((char *) str);
25     return NULL;
26 }
```

例 2 编写一个函数 `is_within()`，它接受两个参数，一个是字符，另一个是字符串指针。其功能是如果字符在字符串中，就返回一个非零值（真）；否则返回 0 值（假）。在一个使用循环语句为这个函数提供输入的完整程序中进行测试。

```
1 #include<stdio.h>
2 #include<stdbool.h>
3 bool is_within(const char * str, int c);
4
5 int main(void) {
6     char * str = "0123456789012345678901234567890";
7     char c = '5';
8
9
10    if (is_within(str, c))
11        printf("Find the character %c\n", c);
12    else
13        printf("Cannot find the character %c!\n", c);
14
15    return 0;
16 }
17
```

```
18 bool is_within(const char * str, int c)
19 {
20
21     while(*str && *str != c)
22         str++;
23     if (*str == c)
24         return true;
25     return false;
26 }
```

例 3 函数 `strncpy(s1, s2, n)` 从 `s2` 复制 `n` 个字符给 `s1`，并在必要时截断 `s2` 或为其填充额外的空字符。如果 `s2` 的长度大于或等于 `n`，目标字符串就没有标识结束的空字符。函数返回 `s1`。自己编写这个函数 `my_strncpy()`，并编制一个完整程序测试它。


```
1 char * strncpy (char * target, const char * source, size_t
count)
2 {
3     char *start = target;
4
5     while (count && (*target++ = *source++))
6         count--;
7
8     if (count)
9         while (--count)
10             *target++ = '\0';
11
12     return (start);
13 }
```

例 4 编写一个函数 `string_in()`，它接受两个字符串指针参数。如果第二个字符串被包含在第一个字符串中，函数就返回被包含的字符串开始的地址。如，`string_in("hats'", "at")` 返回 `hats` 中 `a` 的地址，否则返回 `NULL`。最后编制一个完整程序测试它。

```
1 #include <stdio.h>
```