

# 指针与字符串

## 课后练习题

1. 编写一个函数，用于判断给定字符串是否为回文。函数原型如下：

```
int isPalindrome(char *s);
```

其中，s 为给定字符串。若 s 为回文则函数返回 1，否则函数返回 0。

2. 编写函数实现以下字符串处理功能（不允许调用 `string.h` 中提供的函数）：

(1) 用指针操作实现 C 标准库中的 `strcat` 函数。函数 `strcat(s, t)` 将 t 指向的字符串复制到 s 指向的字符串的尾部。函数原型如下：

```
void strcat(char *s, char *t);
```

说明：假设存储字符串 s 的数组容量充足。

(2) 编写函数 `strend(s, t)`。如果字符串 t 出现在字符串 s 的尾部，该函数返回 1；否则返回 0。函数原型如下：

```
int strend(char *s, char *t);
```

(3) 编写函数 `strindex(s, t)`。该函数返回字符串 t 在 s 中最后一次出现的位置，如果 s 中不包含 t，则返回 -1。函数原型如下：

```
int strindex(char *s, char *t);
```

要求：分别用数组索引和指针操作两种方式实现该函数。

3. 写一个函数，将一个整数转化为十进制表示的字符串，函数原型如下：

```
char* itoa(int n, char *string);
```

其中，n 为待转化的整数，string 为用于存储转化结果的字符串。函数返回字符串 string。

要求：采用递归和非递归两种方法实现该函数。

4. 实现字符串的删除和插入操作。

(1) 写一个函数用于从一个字符串中删除指定位置指定长度的子串，函数原型如下：

```
char* erase(char *string, char *pstr, int n);
```

其中，string 为待处理的字符串，pstr 为指向字符串 string 内部某个字符或者指向末尾空字符的指针，n 为将被删除的子串的最大长度。函数应实现删除字符串 string 从 pstr 所指向的字符开始（包括）往后的至多 n 个字符，并返回字符串 string。

**说明：**如果从 pstr 开始（包括）到字符串的最后一个非空字符（包括）之间有 n 个或 n 个以上字符，则删除从 pstr 指向位置开始的 n 个字符；如果不足 n 个字符，则删除从 pstr 指向位置开始到字符串末尾的所有非空字符。

**应用举例：**

```
1) char s1[20] = "Hello World!";  
   char *ps = s1[6]; //指向字母 'W'  
   erase(s1, ps, 2); //执行后，字符串 s1 的内容为 "Hello rld!"  
2) char s1[20] = "Hello World!";  
   char *ps = s1[6]; //指向字母 'W'  
   erase(s1, ps, 20); //执行后，字符串 s1 的内容为 "Hello "
```

(2) 写一个函数用于将一个字符串插入另一个字符串的指定位置，函数原型如下：

```
char* insert(char* s1, char* s2, char* pos);
```

其中，s1 为指向待插入字符串的指针，s2 为指向目标字符串的指针，pos 为指向字符串 s2 内部的某个字符或末尾的空字符的指针。该函数应实现将字符串 s1 中的所有字符（不包括末尾空字符）插入字符串 s2 中 pos 指向的字符之前的位置，并返回字符串 s2。

**要求：**该函数不得改变字符串 s1。

**说明：**假设用于存储字符串 s2 的数组容量充足。

**应用举例：**

```
char *s1 = "New ";  
char s2[20] = "Hello World!";  
char *ps = s2[6]; //指向字母 'W'  
insert(s1, s2, ps); //执行后，字符串 s2 的内容为 "Hello New World!"
```