

## 1. strlen()

作用:返回字符串s中的字符数(末尾'\0'除外)

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main() {
5     char str[100] = "abc ";
6     printf("%d",strlen(str));
7     return 0;
8 }
9
```

## 2. strcat()

作用:把s2指向的字符串连接到s1指向字符串后面,s2字符串的第一个字符覆盖s1字符串的'\0',该函数返回s1,

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main() {
5     char str1[100] = "abc";
6     char str2[100] = "123";
7     printf("%s",strcat(str1,str2));
8     return 0;
9 }
10
```

## 3. strncat()

作用:把s2前n个字符连接到s1后面,该函数返回s1,

```
1 #include<stdio.h>
2 #include<string.h>
3 int main() {
4     char* s2 = "bbbbbbbbbbbbbbbbbbbb"; //20 'b's
5     char s1[50] = "aaaaaaaaaaaaaaaaaaaaa"; //20 'a's
6     strncat(s1, s2, 10);
7     puts(s1);
8     return 0;
9 }
```

与strncpy不同的是, strncat会自动在末尾加'\0', 若指定长度超过源字符串长度, 则只复制源字符串长度即停止

## 4. strcmp()

作用:比较两个字符串s1和s2,如果s1>s2,返回正数;如果s1<s2,返回负数;如果s1=s2,返回0

```

1  #include<stdio.h>
2  #include<string.h>
3
4  int main() {
5      char str1[100] = "abc";
6      char str2[100] = "abc";
7      char str3[100] = "bcd";
8      if (strcmp(str1, str2) == 0)
9          printf("str1与str2相等\n");
10     if (strcmp(str3, str1) == 1)
11         printf("str3大于str1\n");
12     if (strcmp(str1, str3) == -1)
13         printf("str1小于str3");
14     return 0;
15 }

```

## 5. strncmp()

作用:比较两个字符串s1和s2,如果s1>s2,返回正数;如果s1<s2,返回负数;如果s1=s2,返回0

```

1  #include<stdio.h>
2  #include<string.h>
3
4  int main() {
5      char str1[100] = "abc";
6      char str2[100] = "abc";
7      char str3[100] = "bcd";
8      if (strncmp(str1, str2, 3) == 0)
9          printf("str1与str2相等\n");
10     if (strncmp(str3, str1, 1) == 1)
11         printf("str3与str1相等(前n个字符)\n");
12     if (strncmp(str1, str3, 2) == -1)
13         printf("str1小于str3");
14     return 0;
15 }

```

## 6. strcpy()

作用:把s2指向的字符串拷贝到s1的内存位置(会覆盖s1的内容),返回s1

```

1  #include<stdio.h>
2  #include<string.h>
3
4  int main() {
5      char str1[100] = "abc";
6      char str2[100] = "123";
7      printf("%s", strcpy(str1, str2));
8      return 0;
9  }
10

```

## 7. strncpy()

作用: 将字符串s2中的n个字符拷贝到字符串s1的内存位置(会覆盖s1的内容),返回s1

```
1 #include<stdio.h>
2 #include<string.h>
3 int main() {
4     char* s2 = "bbbbbbbbbbbbbbbbbbbb"; //20 'b's
5     char s1[50] = "aaaaaaaaaaaaaaaaaaaaa"; //20 'a's
6     strncpy(s1, s2, 10);
7     puts(s1);
8     return 0;
9 }
```

## 8. strchr()

作用: 查找字符串s中第一次出现字符c的位置指针, 如果没有找到, 返回NULL

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main() {
5     char str1[100] = "abcdefg";
6     char str2[100] = "123";
7     if (strchr(str1, 'c')) //找到, 则返回指向该字符指针, 输出其后的字符串
8         printf("%s\n", strchr(str1, 'c'));
9     if (strchr(str1, 'n') == NULL) //返回为NULL, 则没有找到
10        printf("未找到! ");
11    return 0;
12 }
13
```

## 9. strstr()

作用: 查找字符串s1中第一次出现字符串s2的位置指针, 如果没有找到, 返回NULL

```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main() {
5     char str1[100] = "abcdefg";
6     if (strstr(str1, "bc")) //找到, 则返回指向该字符指针, 输出该字符串后的所有字符
7         printf("%s\n", strstr(str1, "bc"));
8     if (strstr(str1, "bca") == NULL) //返回为NULL, 则没有找到
9         printf("未找到! ");
10    return 0;
11 }
12
```

## 10. strrev()

```
1 #include<string.h>
2 #include<stdio.h>
3 int main() {
4     char forward[] = "string"; //原文中定义为char*是不对的，指向代码段的指针内容是不可变的
5     printf("Before strrev():%s\n", forward);
6     strrev(forward);
7     printf("Afterstrrev(): %s\n", forward);
8     return 0;
9 }
```