

1.strstr()查找字符串

```
1  #include "head.h"
2
3  //C语言中，strstr()函数用于找到子串在一个字符串中第一次出现的位置,在string.h头文件中。
4  //char *strstr(const char *str1, const char *str2)
5  //str1是总串，str2是需要匹配的字符串位置，返回值为char * 类型。
6  int main()
7  {
8      char buf1[100],buf2[100];
9      scanf("%s",buf1);
10     scanf("%s",buf2);
11
12     char *s=strstr(buf1,buf2);
13     printf("%s\n",s);
14     return 0;
15 }
```

2.strlen()获取字符串长度

```
1  #include "head.h"
2  int main()
3  {
4      //strlen函数返回的是在字符串中'\0'前面出现的字符的个数
5      char buf[100];
6      scanf("%s",buf);
7      printf("%ld\n",strlen(buf));
8      return 0;
9  }
```

3.strcpy()覆盖字符串

```
1  #include "head.h"
2  int main()
3  {
```

```

4 //char *strcpy(char *destination,const char *source)
5 //strcpy是覆盖拷贝，将source全覆盖拷贝到destination，会把'\0'也拷过去，且必须考虑
  destination的空间够不够
6 //(destination的空间必须>=source的空间)
7
8 char buf1[100],buf2[100];
9 scanf("%s",buf1);
10 strcpy(buf2,buf1);
11 printf("%s\n",buf2);
12 return 0;
13 }

```

4.strcat()追加字符串

```

1 #include "head.h"
2 int main()
3 {
4     ////char *strcat(char *destination,const char *source)
5     //strcat追加拷贝,追加到目标空间后面，目标空间必须足够大，能容纳下源字符串的内容
6
7     char buf1[100],buf2[100];
8     scanf("%s",buf1);
9     scanf("%s",buf2);
10
11     strcat(buf1,buf2);
12     printf("%s\n",buf1);
13     return 0;
14 }

```

5.strcmp()比较字符串

```

1 #include "head.h"
2 int main()
3 {
4     //strcmp(const char *s1,const char *s2)
5     //strcmp比较两个字符串的大小，一个字符一个字符比较，按ASCLL码比较

```

```

6 //标准规定:
7 //第一个字符串大于第二个字符串, 则返回大于0的数字
8 //第一个字符串等于第二个字符串, 则返回0
9 //第一个字符串小于第二个字符串, 则返回小于0的数字
10
11 char buf1[100],buf2[100];
12 scanf("%s",buf1);
13 scanf("%s",buf2);
14
15 printf("%d\n",strcmp(buf1,buf2));
16 return 0;
17 }

```

6.strdup()拷贝字符串

```

1 #include "head.h"
2 int main()
3 {
4     //char *strdup(const char *s);
5     /*
6         strdup()函数主要是拷贝字符串s的一个副本, 由函数返回值返回,
7         这个副本有自己的内存空间, 和s没有关联。strdup函数复制一个字符串, 使用完后, 要使用delete函数删除在函数中动态申请的内存,
8         strdup函数的参数不能为NULL, 一旦为NULL, 就会报段错误, 因为该函数包括了strlen函数, 而该函数参数不能是NULL。
9         */
10
11
12
13     char buf1[100];
14     scanf("%s",buf1);
15
16     char *s=strdup(buf1);
17     printf("%s\n",s);
18
19     free(s);
20     return 0;
21 }

```

7.strtok()分割字符串

```
1  #include "head.h"
2  int main()
3  {
4      //第一次使用时第一个参数传入待分割的字符串，第二个参数传入分割符号。
5      //第二次使用时第一个函数传入NULL,第二个参数传入分割符号。
6      char buf[100];
7      char *str;
8      scanf("%s",buf);
9          str=strtok(buf,"1");
10 while(str!=NULL){
11     printf("%s\n",str);
12     str=strtok(NULL,"1");
13 }
14
15 return 0;
16 }
```