

//字符串处理函数内容

//字符串常量的建立

字符串常量是由一对双括号括起来的一个字符序列

为了便于字符串的长度，C编译器会在字符串的末尾加上一个\0

在字符串中不显示的输出，

>>>>>字符串就是一个字符数组，且是一个以\0为结尾的字符数组

>>>>>其的存储下，其的长度会比字符数组再长一个字符'\0'

初始化的情况下，

申请内存，需要对字符数组再多一个字符长度

进行初始化输入的时候需要多一位，多一个输入的'\0'的内容

如果没有'\0'，那么就是一个字符数组，否则就是一个字符数组

必须要大一个字符

/******

字符串的访问和输入和输出

三种方法：

(1) 按照字符进行一个一个的输入

在输出时，当遇到\0作为语句的结

(2) 按照一个整体进行输入

%s进行输入和输出的操作处理

>>>>>用scanf这个函数按照%s格式符不能输入带有空格的字符串内容

(3) 使用字符串处理的函数gets()，可以输入带有空格的字符串

>>>>>gets使用回车键作为字符串的终止符，同时将回车符也从缓冲区中读走，但是不作为字符串中的一部分

>>>>>而scanf不读走回车符，回车符留在缓冲区中

输出的时候使用puts，这个函数可以从括号中给的地址开始，依次输出存储单元中的字符，当遇到第一个'\0'

时进行输出，并自动输出一个换行符。

>>>>>函数gets不能限制输入字符串的长度，很容易引起缓冲区的溢出，给黑客可乘之机

/******

>>>>>转义字符：\ >>>>>当出现双引号或者其他特殊字符时，必须在前面使用转义字符。

>>>>>str1为被操作的str2的作为原有的字符串来进行处理

字符串的处理函数

strlen(str)

>>>>>求长度，计算不包含'\0'在内的所有的字符的个数

strcpy(str1, str2)

>>>>>将str2的内容复制到str1中

strcmp(str1, str2)

>>>>>str1大于str2时为大于0 相等时为 0

strcat(str1, str2)

>>>>>str2添加到str1的后面，str1的最后一个

结束符被覆盖，连接后的字符串放在str1中

strncpy(str1, str2, n)

>>>>>将str2的

的前n个的

内容复制到str1中

strncmp(str1, str2, n)

>>>>> 比较n个字符

str1大于

str2时为大于0 相等时为 0

strncat(str1, str2, n)

>>>>> 添加n个字符

str2添加

到str1的后面，str1的最后一个结束符被覆盖，连接后的字符串放在str1中

同时的有，对单个字符的处理可以使用赋值运算符，而对字符串的处理操作则不可以，需要使用字符串的处理函数的来进行。

/******

向函数传递一个字符串

向函数传递字符串时可以通过 字符数组或 字符指针作为函数的参数。

字符串的循环的结束，通常不使用数字的循环处理来控制，而是使用字符串的结束标志'\0'来判断字符串中的数据是否被遍历结束

使用

for(i = 0; i != '\0'; i++)

do {} while(str[i] != '\0')

```
while(str[i] != '\0')
```

来进行循环读取的控制过程

数组实现字符串的拷贝的过程

数组实现字符串的长度产生的过程

数组实现字符串的对比的过程

```
/******
```

```
/******
```

```
printf("scanf输入:");
```

```
scanf("%s", str);
```

```
printf("gets输入:");
```

```
gets(str2);
```

>>>>>gets输入会直接取得第一次的输入作为gets的输入，请不要将两个输入同时进行使用

>>>>>但是scanf读到第一个空格就不再读下去了，gets则要读到输入者输入回车为止

```
printf("%s\n", str);
```

```
printf("%s\n", str1);
```

```
printf("%s\n", str2);
```

```
/******
```

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

```
int main()
```

```
{
```

```
char str[8];
```

```
char str1[8] = {'a', 'b', 'c', 'd', 'e', 'f'};
```

```
char str2[8];
```

```
printf("scanf输入:");
```

```
scanf("%s", str);
```

```
printf("gets输入:");
```

```

    gets(str2);
    printf("%s\n", str);
    printf("%s\n", str1);
    printf("%s\n", str2);
    return 0;
}

/*****

/*****
    //字符串在使用之前最好初始化一下，否则很容易报各种错
    //以下三种是循环
    for(i = 0;str1[i] != '\0';i++)
    {
        str2[i] = str1[i];
    }
    i = 0;                                //注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化
    do
    {
        str3[i] = str1[i];
        i++;                                //在while和do while循环下，需要自己设置
变量因子来实现循环的完成
    }while(str1[i] != '\0');
    i = 0;                                //注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化
    while(str1[i] != '\0')
    {
        str4[i] = str1[i];
        i++;                                //在while和do while循环下，需要自己设置
变量因子来实现循环的完成
    }

/*****
#include<stdio.h>

```

```

int main()
{
    int i;
    char str1[8] = {'a','b','c','c','e','f','\0'};
    char str2[8] = {'a','b','c','d','e','f','\0'};
    char str3[8] = {'a','b','c','d','e','f','\0'};
    char str4[8] = {'a','b','c','d','e','f','\0'};

    //字符串在使用之前最好初始化一下
    //以下三种是循环
    for(i = 0;str1[i] != '\0';i++)
    {
        str2[i] = str1[i];
    }
    i = 0;                                     //注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化
    do
    {
        str3[i] = str1[i];
        i++;                                  //在while和do while循环下，需要自己设置
变量因子来实现循环的完成
    }while(str1[i] != '\0');
    i = 0;                                     //注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化
    while(str1[i] != '\0')
    {
        str4[i] = str1[i];
        i++;                                  //在while和do while循环下，需要自己设置
变量因子来实现循环的完成
    }

    printf("%s\n",str1);
    printf("%s\n",str2);
    printf("%s\n",str3);

```

```

    printf("%s\n", str4);
    return 0;
}

/*****

//字符串在使用之前最好初始化一下
//以下是字符串函数的使用方法
i = strlen(str1); //>>>>>>求长度，计算不包
含'\0' 在内的所有的字符的个数
strcat(str1, str4); //>>>>>>str2添加到str1的
后面，str1的最后一个结束符被覆盖，连接后的字符串放在str1中
k = strcmp(str1, str3); //>>>>>>str1大于str2时为大于0 相
等时为 0
strcpy(str2, str1); //>>>>>>将str1的内容复制
到str2中

strncpy(str1, str5, n); //>>>>>>将str2的
的前n个的 内容复制到str1中
q = strncmp(str1, str6, n); //相等>>>>>> 比较n个字符
str1大于str2时为大于0 相等时为 0
strncat(str1, str7, n); //>>>>>> 添加n个字符
str2添加到str1的后面，str1的最后一个结束符被覆盖，连接后的字符串放在st
/*****/
#include<stdio.h>
#include<string.h> //使用之前请加一个头文件

int main()
{
    int i; //>>>>>>求长度，计算不包含'\0' 在内的所有的字符的
    个数
    int k; //>>>>>>str1大于str2时为大于0 相等时为 0
    int q; //>>>>>>比较n位，str1大于str2时为大于0 相等时为
    0
    int n = 5; //n类型的字符处理

```

```
char str1[16] = {'c','c','c','c','c','c','\0'};
```

```
char str2[16] = {'a','b','c','d','e','f','\0'};
```

```
char str3[16] = {'a','f','c','d','e','f','\0'};
```

```
char str4[16] = {'a','b','c','d','e','f','\0'};
```

```
char str5[16] = {'a','b','c','d','e','f','\0'};
```

```
char str6[16] = {'a','b','c','d','e','f','\0'};
```

```
char str7[16] = {'a','b','c','d','e','f','\0'};
```

//字符串在使用之前最好初始化一下

//以下是字符串函数的使用方法

```
i = strlen(str1); //>>>>>>求长度，计算不包
```

含'\0'在内的所有的字符的个数

```
strcat(str1, str4); //>>>>>>str2添加到str1的
```

后面，str1的最后一个结束符被覆盖，连接后的字符串放在str1中

```
k = strcmp(str1, str3); //>>>>>>str1大于str2时为大于0 相
```

等时为 0

```
strcpy(str2, str1); //>>>>>>将str1的内容复制
```

到str2中

```
strncpy(str1, str5, n); //>>>>>>将str2的
```

的前n个的内容复制到str1中

```
q = strncmp(str1, str6, n); //相等>>>>>> 比较n个字符
```

str1大于str2时为大于0 相等时为 0

```
strncat(str1, str7, n); //>>>>>> 添加n个字符
```

str2添加到str1的后面，str1的最后一个结束符被覆盖，连接后的字符串放在st

```
printf("长度: %d\n", i);
```

```
printf("不相等: %d\n", k);
```

```
printf("相等: %d\n", q);
```

```
printf("%s\n", str1);
```

```
printf("%s\n", str2);
```

```

printf("%s\n", str3);
printf("%s\n", str4);
printf("%s\n", str5);
printf("%s\n", str6);
printf("%s\n", str7);

return 0;
}

/*****
//字符串在使用之前最好初始化一下，否则很容易报各种错
//以下三种是循环
for(i = 0; str1[i] != '\0'; i++)
{
    str2[i] = str1[i];
}
i = 0;                                //注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化
do
{
    str3[i] = str1[i];
    i++;                                //在while和do while循环下，需要自己设置
变量因子来实现循环的完成
}while(str1[i] != '\0');
i = 0;                                //注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化
while(str1[i] != '\0')
{
    str4[i] = str1[i];
    i++;                                //在while和do while循环下，需要自己设置
变量因子来实现循环的完成
}

*****/
#include<stdio.h>

```



```

int main()
{
    int i;
    int len;
    int choice = 0;

    char str1[8] = {'a','b','c','c','e','f','\0'};
    char str2[8] = {'a','c','c','d','e','f','\0'};
    char str3[8] = {'a','b','c','d','e','f','\0'};
    char str4[8] = {'a','c','c','d','e','f','\0'};

    //数组实现字符串的拷贝的过程
    i = 0;
    for(i = 0;str1[i] != '\0';i++)
    {
        str2[i] = str1[i];
    }

    //数组实现字符串的对比的过程
    i = 0;                                     //注意使用一个变量完成一次的循环工作时，
    实现下一次的循环之前需要将其进行初始化
    while(str1[i] != '\0')
    {
        if(str4[i] == str1[i])
        {
            choice = 0;
        }
        else if(str4[i] <= str1[i])
        {
            choice = -1;
            break;
        }
        else
        {

```

```

        choice = 1;
        break;
    }
    i++;
    //在while和do while循环下，需要自己设置
    变量因子来实现循环的完成
}

```

```

//数组实现字符串的长度产生的过程
i = 0;
//注意使用一个变量完成一次的循环工作时，
实现下一次的循环之前需要将其进行初始化

```

```

len = 0;
do
{
    i++;
    len = len+1;
    //在while和do while循环下，需要
    自己设置变量因子来实现循环的完成
}while(str1[i] != '\0');

```

```

printf("%s\n", str1);
printf("%s\n", str2);
printf("长度: %d\n", len);
printf("不相等: %d\n", choice);
return 0;
}

```

```

/*****

```

```

//字符串转换函数的使用
atof//转换为双精度浮点数
atoi//转换为整数
atol//转换为长整数
//这段代码任然存在问题

```

```

/*****

```

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

int main()
{
    int intNum;
    long longNum;
    double doubleNum;
    char str[8];
    str[8] ="14.55";

    doubleNum = atof(str);
    intNum = atoi(str);
    longNum = atol(str);

    printf("doubleNum = %d\n", doubleNum);
    printf("intNum = %f\n", intNum);
    printf("longNum = %ld\n", longNum);
    return 0;
}
*/
```