

## 程序设计基础 (C) ——第6章数组

郑州大学软件学院/网络空间安全学院

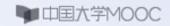
↓ Lecturer: 宋轩

Office : 行政楼-306

Email : songxuan@zzu.edu.cn

# 第6章 数组——字符数组

#### 字符数组的定义、引用和初始化



- ◆ 字符数组的定义方法和普通数组的定义方式相同
  - ◎ 例如:char c[10];
- ▼ 对字符数组也可以采用普通数组的初始化方式
  - ◎ 例如:char c[5]={'H','e','I','I','o'};
- ◆ 引用字符数组中的元素类似
  - ◎ 例如:c[i] 像使用普通字符变量一样使用

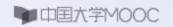
#### 字符数组的定义、引用和初始化



- ◆ 字符数组的定义方法和普通数组的定义方式相同
  - ◎ 例如:char c[10];
- ◆ 对字符数组也可以采用普通数组的初始化方式
  - ◎ 例如: char c[5]={'H','e','I','I','o'};
- ◆ 引用字符数组中的元素类似
  - ◎ 例如:c[i] 像使用普通字符变量一样使用

```
//字符数组按逐个字符输出
#include<stdio.h>
int main()
   char c[5]={'H','e','l','l','o'};
   int i:
   for(i=0;i<5;i++)
      printf("%c",c[i]);
    printf("\n");
    return 0;
```

### 字符串



◆ 字符串常量:用一对双引号括起的字符序列

◎ 例如:"hello" "How are you!"

◆ 在存储字符串时,自动在其后添加空字符 '\0',作为字符串结束符。



◆ 注意:空字符不同于空格字符



### 字符串



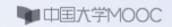
- ◆ C语言没有提供专门的字符串变量,使用字符数组来存储字符串
- ◆ 字符数组:即每个元素都是字符类型的数组

例如: char str[80];

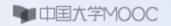
Н	0	w	а	r	е	у	0	u	!	/0
Н	0	w	а	r	е	у	o	u	!	

- ◆ 字符数组元素中必须有 '\0' 才表示存储的是字符串
- ◆ 否则就是若干个普通字符

#### 用字符数组处理字符串



- ♦ 字符数组的初始化
  - 用字符常量的初始化列表对数组初始化char str[6] = {'C','h','i','n','a','\0'};
  - 用字符串常量直接对数组初始化char str[6] = {"China"};
    - char str[6] = "China";
      char str[] = "China";
- ◆ 使用时可以逐个元素处理,也可以用整体处理。



♦ 按数组元素逐个处理每个字符

```
char str[10] = "China";
```

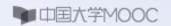
```
for(i=0; str[i] != '\0'; i++)

putchar(str[i]);

putchar('\n');
```

使用串结束符 '\0' 控制循环, 而不是数组的长度,如i<10





▶ 按数组元素逐个处理每个字符 char str[10] = "China"; 等价于 str[i]!=0 或者 str[i]

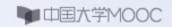
```
for(i=0; str[i] != '\0' ; i++)

putchar(str[i]);

putchar( '\n' );
```

使用串结束符 '\0' 控制循环, 而不是数组的长度, 如i<10

要区别字符串的长度和存放字 符串的字符数组的长度。 字符串 "China" 的长度是5, 而字符数组str的长度是10。

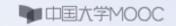


◆ 按字符串整体输入/输出 char str[80];

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

gets(str);
puts(str);





♦ 按字符串整体输入/输出 char str[80]; 不能输入带空格的字符串 scanf("%s",str); printf("%s",str); 可以输入带空格的字符串 gets(str); puts(str);



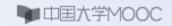
◆ 按字符串整体输入/输出 char str[80];

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

```
gets(str);
puts(str);
```

```
#include < stdio.h >
int main()
{
    char str[80];
    scanf( "%s" ,str);
    printf( "%s" ,str);
    return 0;
}
```

```
当输入为
how are you
输出为
how
```



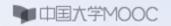
♦ 按字符串整体输入/输出 char str[80];

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

```
gets(str);
puts(str);
```

```
#include < stdio.h >
int main()
{
    char str[80];
    gets(str);
    puts(str);
    return 0;
}
```

```
当输入为
how are you
输出为
how are you
```

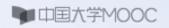


输入一个句子(一行字符),统计其中有多少个单词。句子中各单词之间用一个以 上的空格分隔开。

#### 思路是:

- 1、定义存放句子的字符数组及相应变量;
- 2、输入句子;
- 3、统计单词个数;
- 4、输出单词个数。





输入一个句子(一行字符),统计其中有多少个单词。句子中各单词之间用一个以 上的空格分隔开。

#### 思路是:

#### 统计单词过程:

从串首开始直到字符串结束,依次检查字符串各字符;如果相邻两字符,前者为空格后者为非空格,说明后者是某单词的开始字符,则单词数量加1;如果句子不是以空格开始,句子的第一个单词无法统计,这时单词数初值应该是1。





输入一个句子(一行字符),统计其中有多少个单词。句子中各单词之间用一个以

上的空格分隔开。

思路是:

#### 统计单词过程:

从串首开始直到字符串结束,依次检查字符串各字符; 如果相邻两字符,前者为空格后者为非空格,说明后 者是某单词的开始字符,则单词数量加1; 如果句子不是以空格开始,句子的第一个单词无法统 计,这时单词数初值应该是1。

```
#include <stdio.h>
int main()
    char str[255];
    int i,num =0;
    gets(str);
    if(str[0]!='') num=1;
    for(i=0;str[i];i++)
        if(str[i]==' '&&str[i+1]!=' ')
                  num++;
    printf( "words : %d\n",num);
    return 0:
```

As a second of the second



输入一个句子(一行字符),统计其中有多少个单词。句子中各单词之间用一个以

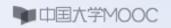
上的空格分隔开。

思路是:

#### 统计单词过程:

从串首开始直到字符串结束,依次检查字符串各字符;如果相邻两字符,前者为空格后者为非空格,说明后者是某单词的开始字符,则单词数量加1;如果句子不是以空格开始,句子的第一个单词无法统计,这时单词数初值应该是1。

```
#include <stdio.h>
int main()
    char str[255];
   int i,num=0;
    gets(str);
   tor(i=0;str|i|;i++)
       if(str[i]==' '&&str[i+1]!=' ')
                num++;
    printf( "words : %d\n",num);
    return 0;
```



连接两个串:把字符串str2连接到字符串str1之后。如果字符数组str1的存储空间不够, 直到把str1填满为止。

#### 思路是:

- 1、定义存放两个串的字符数组及相关变量;
- 2、输入两个串;
- 3、定位串1的串结束符;
- 4、依次将串2的每个字符连接到串1后,直至串2处理完或者存放串1的数组空间不够;
- 5、在连接好后,末尾填入字符串结束符;
- 6、输出连接后的串。

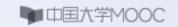
```
#include<stdio.h>
int main()
    char str1[10], str2[10];
    int i=0, j=0;
    gets(str1); gets(str2);
    while(str1[i]) i++;
    while(str2[j]&& i < 9)
         str1[i]=str2[j];
         i++; j++; }
    str1[i] = '\0';
    puts(str1);
    return 0;
```

#### 字符串处理函数

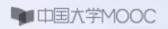


```
#include<string.h>
strlen(字符串); //求字符串的长度, string length
strcpy(目的字符串, 源字符串); // 串复制, string copy
strcat(目的字符串, 源字符串); // 串连接, string combination
strcmp(字符串1, 字符串2); //串比较, string comparison
```

### 字符串处理函数



函数功能	函数调用的一般形式	功能描述及其说明					
求字符串长度	strlen(str);	由函数值返回字符串 str 的实际长度,即不包括\0'在内的实际字符的长度					
字符串拷贝	strcpy(str1,str2);	将字符串 str2 复制到字符数组 strl 中,这里应确保字符数组 strl 的 大小足以存放得下字符串 2					
字符串比较	strcmp(str1,str2);	比较字符串 strl 和字符串 str2 的大小,结果分为 3 种情况:  · 当 strl 大于 str2 时,函数返回值大于 0  · 当 strl 等于 str2 时,函数返回值等于 0  · 当 strl 小于 str2 时,函数返回值小于 0  字符串的比较方法为:对两个字符串从左至右按字符的 ASCII 码值大小逐个字符相比较,直到出现不同的字符或遇到\0'为止					
字符串连接	strcat(str1,str2);	将字符串 str2 添加到字符数组 str1 中的字符串的末尾,字符数组 str1 中的字符串结束符被字符串 str2 的第一个字符覆盖,连接后的字符串存放在字符数组 str1 中,函数调用后返回字符数组 str1 的首地址。这里,字符数组 str1 应定义得足够大,以便能存放连接后的字符串					



从键盘输入三个英文单词,输出其中最大的单词(单词大小按字典顺序)。

思路是:

定义存放三个单词及最大单词的字符数组:

char s1[15],s2[15],s3[15],max[15];

输入三个单词;

前两个单词进行比较,将较大者赋值给max;

将max与第三个单词进行比较,确定最大单词存放在max中;

输出最大的单词

```
#include < stdio.h >
#include<string.h>
int main()
    char s1[15],s2[15],s3[15],max[15];
    gets(s1); gets(s2); gets(s3);
    if(strcmp(s1,s2)>0)
         strcpy(max,s1);
    else
         strcpy(max,s2);
    if(strcmp(s3,max)>0)
         strcpy(max,s3);
    puts(max);
    return 0;
```

# Questions & Answers