

字符数组

ARRAY STRUCTURE

4 数组与结构



10周年庆典

寻找James



寻找James

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
    const int n = 20,m=5;
    char name[][n] = {
        "Kate.Wate","James.Tan","Bull.Ben","Jimes.Tid
        e","James.Ting" };
    char James[] = "James";
    int i,j,len=strlen("James");
    for (i = 0;i < m;++i){
        for (j = 0;j < len;++j){
            if (name[i][j] != James[j])break;
        }
        if(j==len)
            printf("%s has James\n", name[i]);
    }
    system("pause");
    return 1;
}
```

寻找James

用二维的字符数组存储所有会员姓名，然后依次查找姓名含有James的人，打印全名。

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
    const int n = 20,m=5;
    char name[n][m] = {
        "James.Tan","Bull.Ben","Jimes.Tid",
        "James",
        "James");
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if(name[i][j] != James[j])break;
        }
        if(j==len)
            printf("%s has James\n", name[i]);
    }
    system("pause");
    return 1;
}
```



寻找James

一维字符数组James存储James常量。

```
char James[] = "James";
```

字符数组

存放字符的数组

每个元素为一个字符

一个一维数组可以存放一个字符串

寻找James

一维字符数组James存储James常量.

```
char James[] = "James";
```

存放字符的数组

每个元素为一个字符

一个一维数组可以存放一个字符串



寻找James

用二维的字符数组存储
所有会员姓名

```
char name[][n] = {  
    "Kate.Wate", "James.Tan", "Bull.Ben", "  
    Jimes.Tide", "James.Ting" };
```

- 一个二维数组可以存放多个字符串

这里存放5个字符串

name[0]为存放第1个字符串的一维数组名,
name[4]为存放第5(最后)个字符串的一维数组名,
name[0][0]为第1个字符串的第一个字符.



寻找James

求字符串长度。

```
len=strlen("James");
```

字符串处理函数，需要包括头文件string.h

字符数组定义

```
char str[10];
```

字符数组初始化

- `char str[10];`
- 可逐个元素赋值:
`str[0]= 'c' ;str[1]= 'h' ;str[2]= 'i' ;`
`str[3]= 'n' ; str[4]= 'a' ;`

字符数组初始化

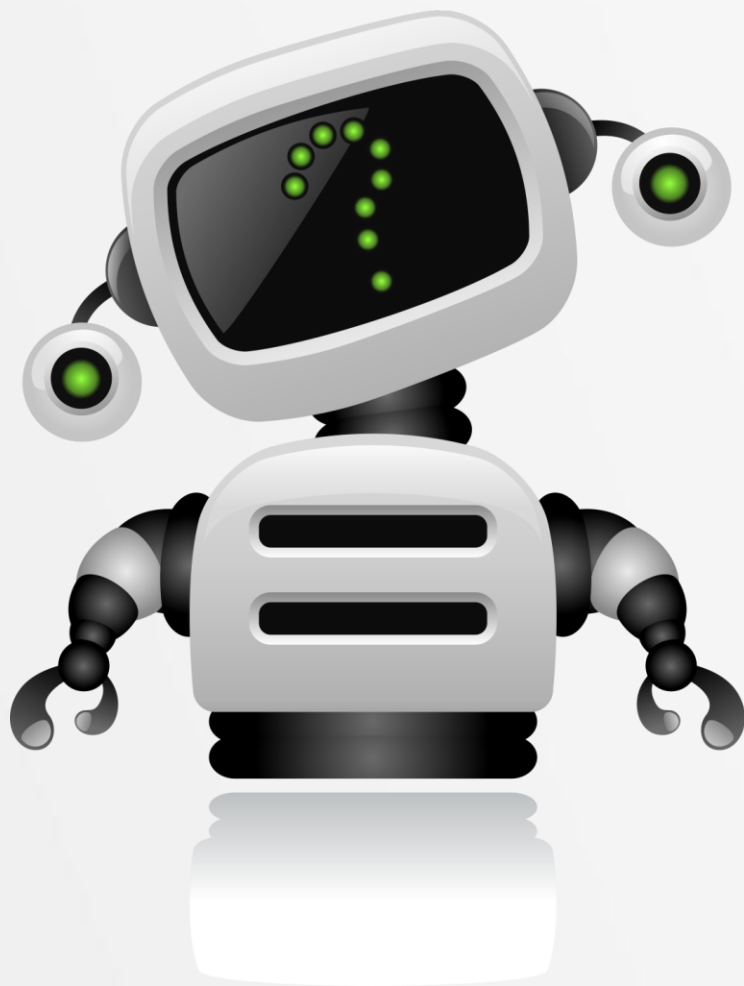
- char str[10];
- 可逐个元素赋值
- 可定义时初始化:

用字符串常量初始化时,自动加结束标志\0.

- `char str[10]={ 'c' , 'h' , 'i' , 'n' , 'a' };`
- `char s1[20]={ " How do you do "};`
- `char s1[20]=" How do you do" ;`



定义字符数组后，还有哪些初始化方法？





4.3.2 字符数组的输入输出

1. 单个字符输入输出

```
char str[10];  
for(i=0;i<5;i++)  
    scanf(" %c" ,&str[i]);  
for(i=4;i>=0;i--)  
    printf(" %c" ,str[i]);
```



4.3.2 字符数组的输入输出

1. 单个字符输入输出
2. 字符串整体输入输出

```
char str[10];  
scanf(" %s" ,str); /*数组名str是数组的起始地址，  
不用&str */  
printf(" %s" , str);
```



4.3.2 字符数组的输入输出

1. 单个字符输入输出
2. 字符串整体输入输出
3. gets函数
4. puts函数

格式为

`puts(字符数组名);`

其作用是将一个字符串输出到终端，并在输出时将字符串结束标记'\0'转换则'\n'，即输出完字符串后换行。如

```
char word[]="abc", word1[]="def";  
puts(word); puts(word1);
```




字符串函数 简介

1.字符串连接函数strcat

调用形式:

strcat(字符数组1,
字符数组2);

功能:将字符数组2连接
于字符数组1的后面.

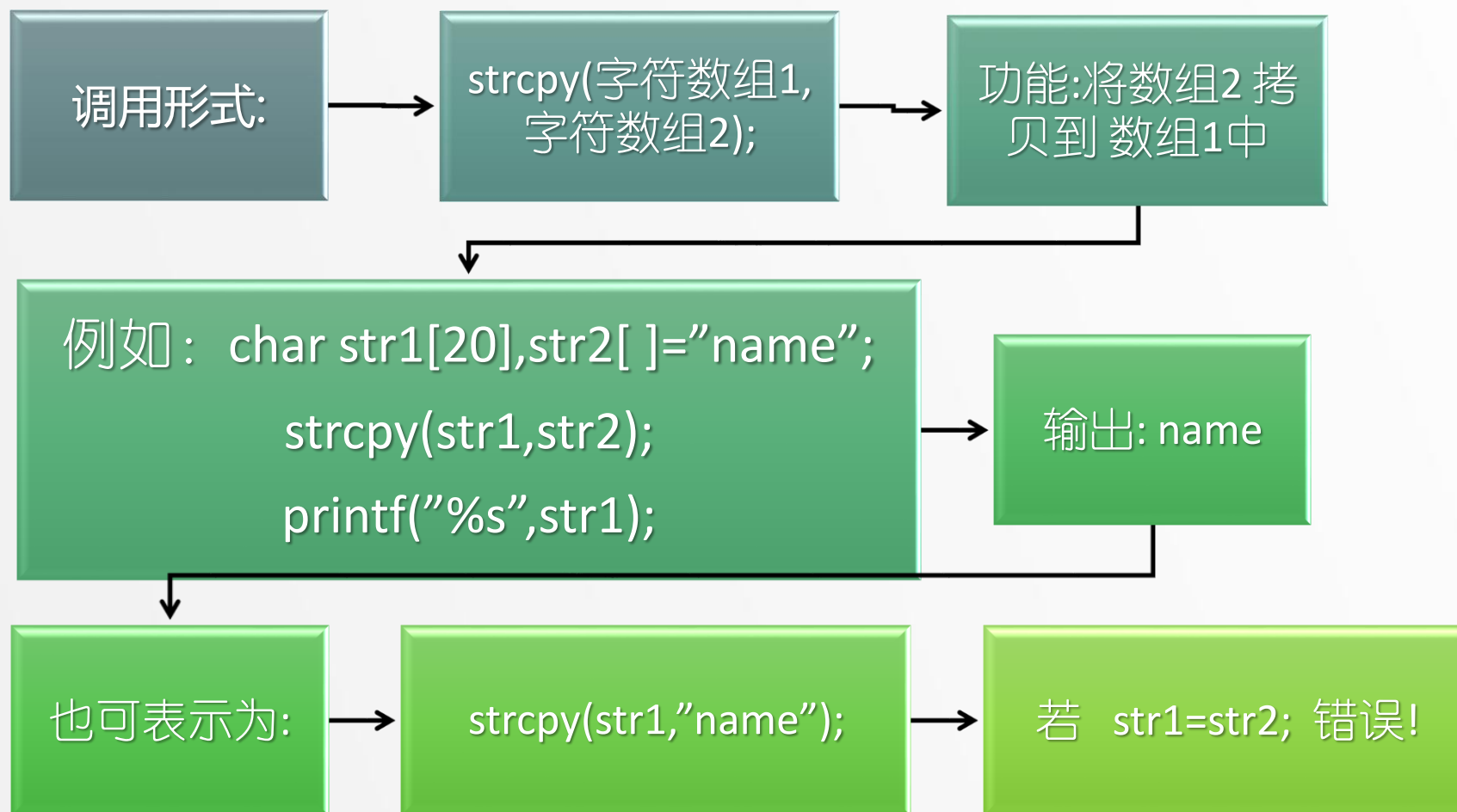
例如:

- char str1[20]="This is a";
- char str2[]="string.";
- printf("%s",strcat(str1,str2));

输出:

This is astring.

2.字符串拷贝函数strcpy



3. 字符比较函数strcmp

调用形式:

`strcmp(字符串1, 字符串2);`

功能:

将两字符串的字符从左到右逐个进行比较. 返回值为比较结果.

字符串1=字符串2 返回值为0
字符串1>字符串2 返回值为正数
字符串1<字符串2 返回值为负数

例如:

`i=strcmp("chinese","china");`

则: i等于1

4.测试字符串长度函数strlen

调用形式:

`strlen(字符数组);`

功能: 测试字符数组
中字符串的实际长度.

输出: 7

例如:

```
char str1[10]={ "Turbo c"};  
printf("%d", strlen(str1));
```

编程练习

如果采用字符串处理函数，怎么求所有校友名字包含 James 的那些人？，比如 “K.James.T” 也应该是邀请人

