第6章 数组（1）

复习回顾

- 上次课的内容：
  - 循环嵌套举例
  - break
  - Continue
  - 一维数组
  - 上机课编程的时候一定要淡定
    - 别抓狂，你抱怨电脑的同时也许电脑也在抱怨你

怎样定义二维数组

- 二维数组定义的一般形式为

  类型符 数组名[常量表达式][常量表达式];

  如：float a[3][4],b[5][10];

- 二维数组可被看作是一种特殊的一维数组：

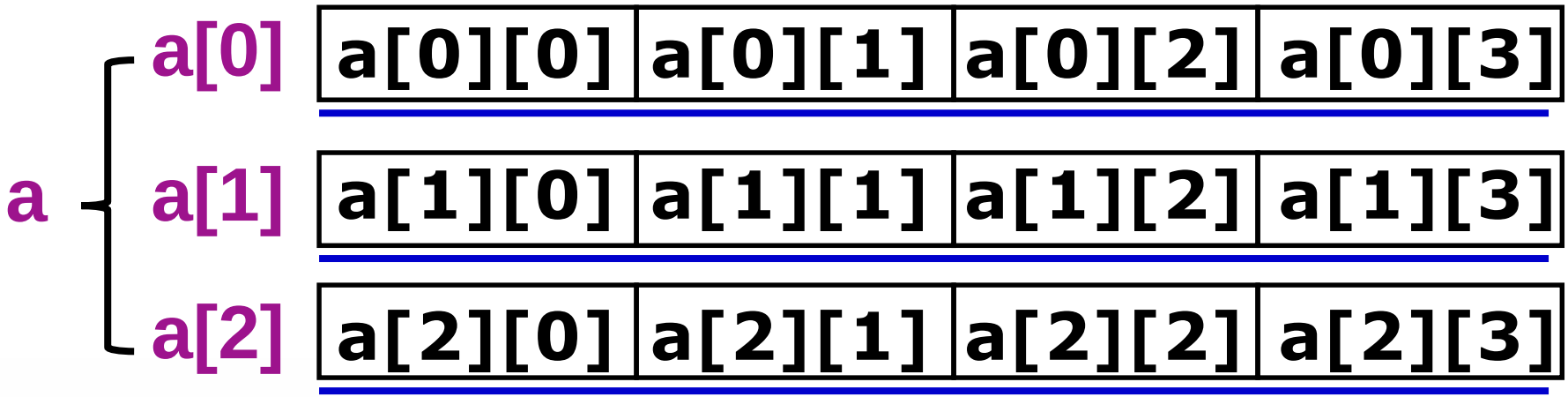
  它的元素又是一个一维数组

怎样看待二维数组

- 例如，float a[3][4]，把a看作是一个一维数组，它有3个元素：

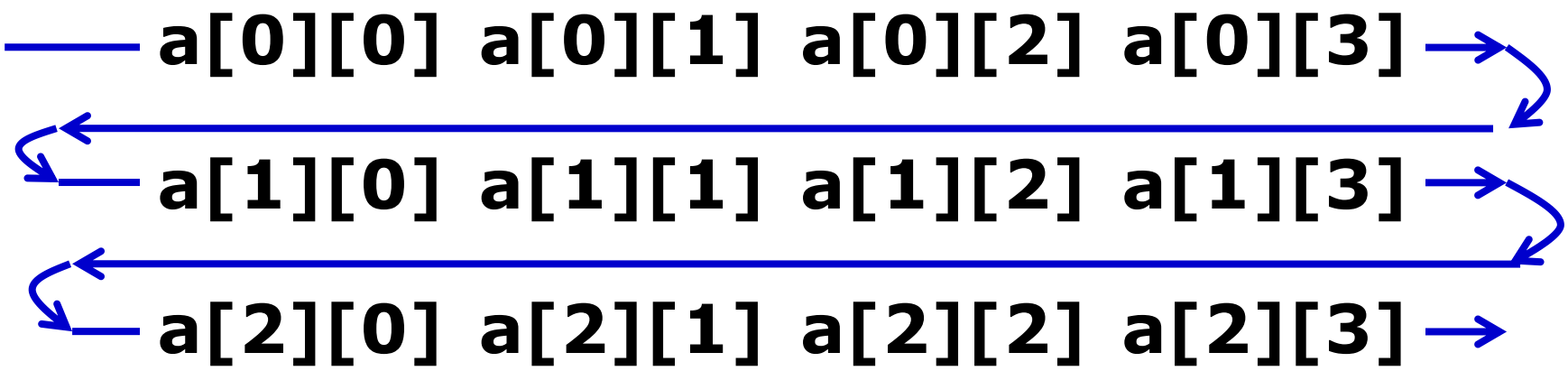
  a[0]、a[1]、a[2]

  每个元素又是一个包含4个元素的一维数组



二维数组的存储

- 行优先，在内存中依次存放：



内存中的存储顺序

怎样引用二维数组中的元素

- 引用二维数组元素的表示形式为：

  数组名 [下标] [下标]

  没有空格　　没有空格

  - 例如，b[1][2]=a[2][3]/2　　合法
  - for(i=0;i<m;i++)

    printf("%d,%d\n",a[i][0],a[0][i]);合法

二维数组的初始化

```
int a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

等价于 （下面的可读性差，我们推荐上面的）

```
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

```
int a[3][4]={{1},{5}};
```

等价于 （可选择性地对部分数组元素赋值）

```
int a[3][4]={{1,0,0,0},{5,0,0,0},{0,0,0,0}};
```

```
int a[3][4]={{1},{},{5,6}};
```

等价于 （可以跳过其中的某行，对其他行进行赋值）

```
int a[3][4]={{1},{0},{5,6}};
```

在二维数组初始化时偷懒

如果初始化时对二维数组的全部元素进行了赋值，则可以省略第一维的长度，但第二维的长度不能省略。

```
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

等价于：

```
int a[ ][4]={1,2,3,4,5,6,7,8,9,10,11,12};
int a[][4]={{0,0,3},{ },{0,10}};
```

合法

二维数组程序举例：转置

- 将一个二维数组行和列的元素互换，存到另一个二维数组中。

$$a=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \Longrightarrow b=\begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix}$$

- 解题思路：
  - 可以定义两个数组：数组a为2行3列，存放指定的6个数，数组b为3行2列，开始时未赋值
  - 将a数组中的a[i][j]存放到b数组中的b[j][i]元素中
  - 用嵌套的for循环完成

二维数组程序举例：转置

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a[2][3]={{1,2,3},{4,5,6}}, b[3][2], i, j;
5.      printf("array a:\n");
6.      for (i=0;i<=1;i++)
7.      {
8.          for (j=0;j<=2;j++)
9.          {
10.             printf("%5d",a[i][j]);
11.             b[j][i]=a[i][j];
12.         }
13.         printf("\n");
14.     }
15.     printf("array b:\n");
16.     for (i=0;i<=2;i++)
17.     {
18.         for (j=0;j<=1;j++)
19.             printf("%5d",b[i][j]);
20.         printf("\n");
21.     }
22.     return 0;
23.}
```

```
array a:
    1    2    3
    4    5    6
array b:
    1    4
    2    5
    3    6
```

二维数组程序举例：最大值

- 有一个3×4的矩阵，要求编程序求出其中值最大的那个元素的值，以及其所在的行号和列号。
- 解题思路：采用“打擂台算法”
  - 先找出任一人站在台上，第2人上去与之比武，胜者留在台上
  - 第3人与台上的人比武，胜者留台上，败者下台
  - 以后每一个人都是与当时留在台上的人比武，直到所有人都上台比为止，最后留在台上的是冠军

二维数组程序举例：最大值

```
……
1.   int i, j, row=0, colum=0, max;
2.   int a[3][4]={{1,2,3,4},{9,8,7,6},{-10,10,-5,2}};
3.   max = a[0][0];
4.   for (i=0; i<=2; i++)
5.       for (j=0; j<=3; j++)
6.           if (a[i][j] > max)
7.           {
8.               max = a[i][j];
9.               row = i;
10.              colum = j;
11.          }
12.  printf("max=%d\nrow=%d\ncolum=%d\n",max,row,colum);
……
```

记最大值

记行号

记列号

```
max=10
row=2
colum=1
```

字符数组的基本概念

- 文字是一种最常见的信息记录方式，由此可见存储字符的重要性
- 用来**存放字符数据**的数组是字符数组
- 字符数组中的**一个元素**存放**一个字符**
- 定义字符数组的方法与定义数值型数组的方法类似

字符数组的定义

```
char c[10];
c[0]='I';   c[1]=' ';
c[2]='a';   c[3]='m';
c[4]=' ';   c[5]='h';
c[6]='a';   c[7]='p';
c[8]='p';   c[9]='y';
```

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
I		a	m		h	a	p	p	y

字符数组的初始化

```
char c[10]={'I',' ','a','m',' ','h','a','p','p','y'};
```

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
I		a	m		h	a	p	p	y

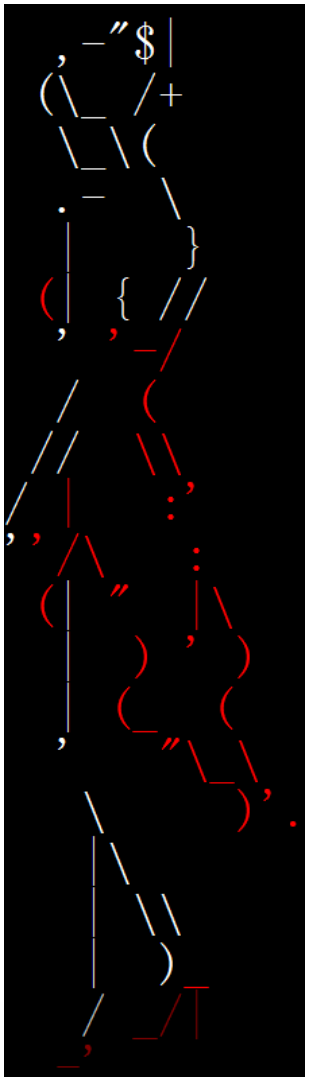
```
char c[10]={'c',' ','p','r','o','g','r','a','m'};
```

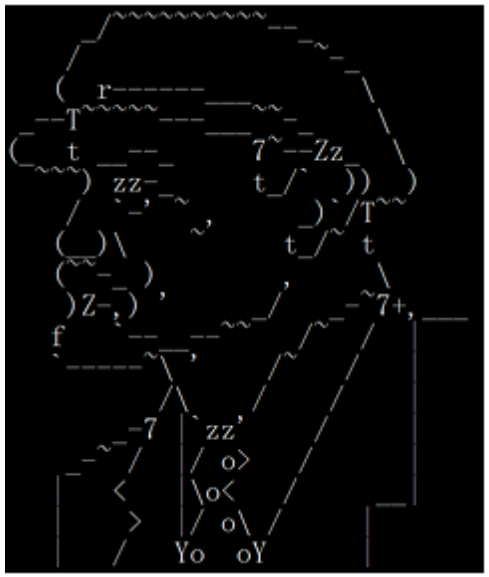
c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
c		p	r	o	g	r	a	m	\0

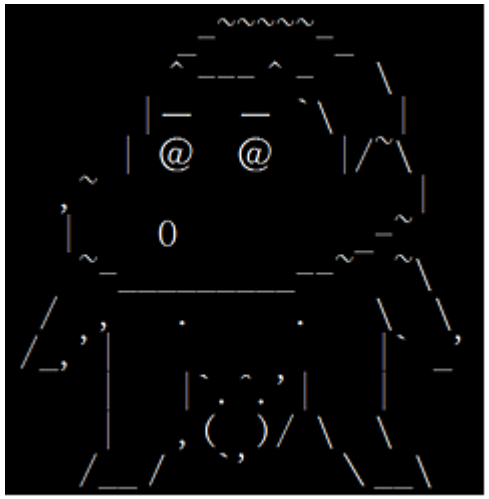
二维字符数组的初始化

```
char diamond[5][5]={{' ',' ','*'},
                    {' ','*',' ','*'},
                    {'*',' ',' ',' ','*'},
                    {' ','*',' ','*'},
                    {' ',' ','*'}};
```

ASCII字符画







引用字符数组元素举例：输出

- 输出一个已知的字符串。
- 解题思路：
  - 定义一个字符数组，并用“初始化列表”对其赋以初值
  - 用循环逐个输出此字符数组中的字符

引用字符数组元素举例：输出

```
1. #include <stdio.h>
2. int main()
3. {
4.     char c[15]={'I',' ','a','m',' ','a',
                ' ','s','t','u','d','e','n','t','.'};
5.     int i;
6.     for (i=0; i<15; i++)
7.         printf("%c",c[i]);
8.     printf("\n");
9.
10.    return 0;
11.}
```

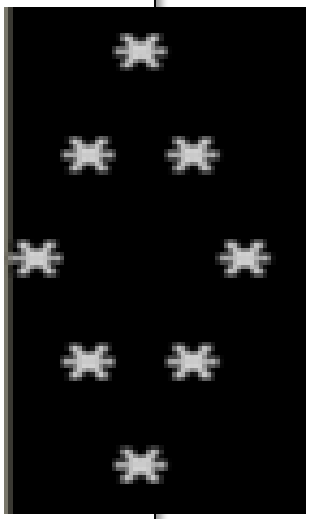
```
I am a student.
```

引用二维字符数组元素：菱形

- 输出一个菱形图。
- 解题思路：
  - 定义一个字符型的二维数组，用“初始化列表”进行初始化
  - 用嵌套的for循环输出字符数组中的所有元素。

引用二维字符数组元素：菱形

```
1. #include <stdio.h>
2. int main()
3. {
4.      char diamond[][5]={{' ',' ','*'},
5.                         {' ','*',' ','*'},
6.                         {'*',' ',' ',' ','*'},
7.                         {' ','*',' ','*'},
8.                         {' ',' ','*'}};
9.      int i,j;
10.     for (i=0;i<5;i++)
11.     {
12.         for (j=0;j<5;j++)
13.             printf("%c",diamond[i][j]);
14.         printf("\n");
15.     }
16.     return 0;
17.}
```



字符串

- 在C语言中，是将字符串作为字符数组来处理的
- 关心的是字符串的有效长度而不是字符数组的长度，通常这两者并不相等！
- 为了测定字符串的实际长度，C语言规定了字符串结束标志'\0'

字符串结束标志'\0'

- '\0'代表ASCII码为0的字符，注意和字符'0'之间的区别！
- 从ASCII码表可以查到，ASCII码为0的字符不是一个可以显示的字符，而是一个“空操作符”，即它什么也不做
- 用它作为字符串结束标志不会产生附加的操作或增加有效字符，只起一个供辨别的标志

用字符串常量初始化字符数组

```
char c[]={"I  am  happy"};
```

可写成

```
char c[]="I  am  happy";
```

相当于

```
char c[11]={"I  am  happy"};
```

注意：字符串实际长度是10，但是还需要存储一个结束标志

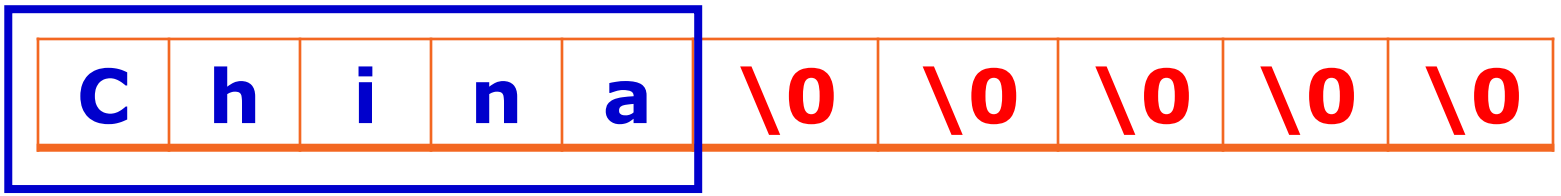
定长字符数组的初始化

char c[10]={"China"};

可写成

char c[10]="China";

从c[5]开始，元素值均为\0



printf(”%s”,c);只显示蓝色方框部分

字符数组输入输出方法

- 两种方法：
  - 逐个字符输入输出（%c）

```
char a[10]; int i,n=-1;
do {
    scanf("%c",&a[++n]);
} while (a[n]!='\n');
```

```
for (i=0; i<n; i++)
{
    printf("%c", a[i]);
}
printf("\n");
```

  - 整个字符串一次输入输出（%s）

```
char a[10];
scanf("%s", &a);//或scanf("%s",a)，注意！这里的a是地址
printf("%s", a);
```

字符数组输入输出注意事项

- printf结合%s输出的字符中不包括结束符'\0'
- 用%s输出字符串时，printf函数中的输出项是字符数组名，不是数组元素名
- 如果一个字符数组中包含多个'\0'，则遇第一个'\0'时输出就结束
- 可以用scanf函数结合%s输入一个字符串，scanf函数中的输入项c是已定义的字符数组名，输入的字符串应短于已定义的字符数组的长度

单个%s输入举例

```
char c[6];
scanf("%s",c);   China↙
```

OK，系统自动在China后面加一个'\0'

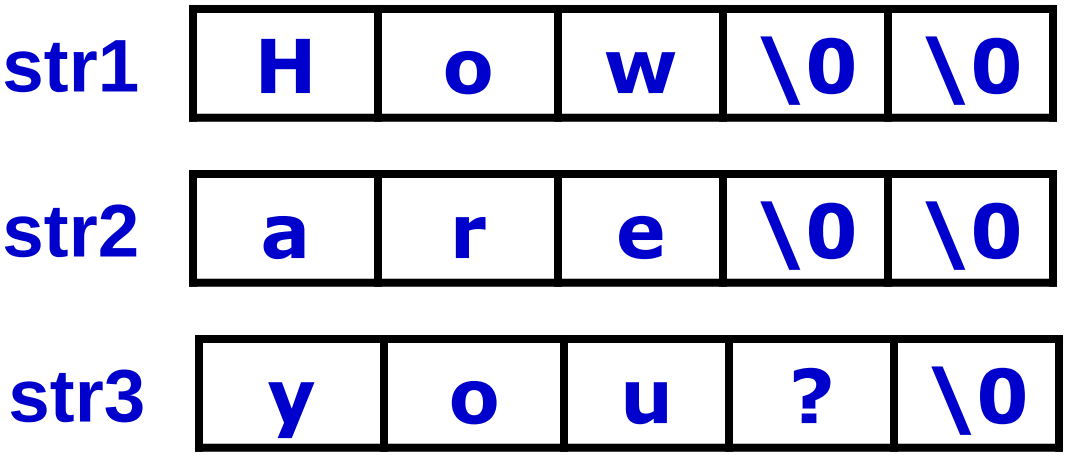
```
scanf("%s",c);   Xiamen↙
```

不妙，后果难以预料...

多个%s输入举例

```
char str1[5],str2[5],str3[5];
scanf("%s%s%s",str1,str2,str3);
```

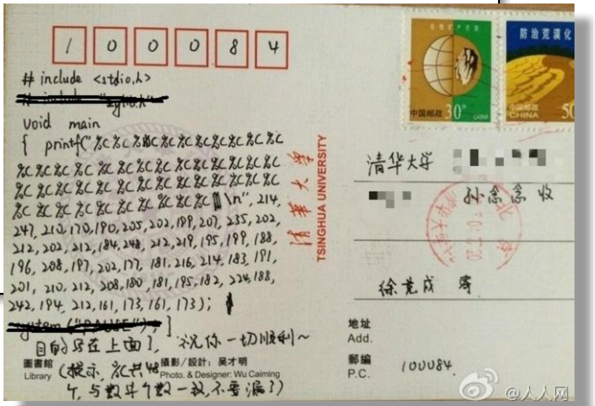
How are you? ↙



老坑：明信片里的奥秘

```c
#include <stdio.h>
int main()
{
    printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",214,247,210,170,190,205,202,199,207,235,202,212,202,212,184,248,212,219,195,199,188,196,208,197,202,177,181,216,214,183,191,201,210,212,208,180,181,195,182,224,188,242,194,212,161,173,161,173);

    return 0;
}
```



这些神秘的整数怎么来的？

学了字符数组，能不能通过编程揭开这个谜题？

字符串处理函数puts

使用字符串函数时,在程序开头加#include <string.h>

1.puts函数----输出字符串的函数

- 其一般形式为：

  puts (字符数组)

- 作用是将一个字符串输出到终端

```
char str[20]="China";
puts(str);
```

输出China

字符串处理函数gets

使用字符串函数时,在程序开头加#include <string.h>

2. gets函数----输入字符串的函数

- 其一般形式为：

  gets(字符数组)

- 作用是输入一个字符串到字符数组

```
char str[5];
gets(str);
Hi↙
```

str

H	i	\0	\0	\0

字符串处理函数strcat

使用字符串函数时,在程序开头加#include <string.h>

3. strcat函数----字符串连接函数

- 其一般形式为：

  strcat(字符数组1，字符数组2)

- 其作用是把两个字符串连接起来，把字符串2接到字符串1的后面，结果放在字符数组1中

strcat函数使用举例

使用字符串函数时,在程序开头加#include <string.h>

3. strcat函数----字符串连接函数

```
char str1[30]="People";
char str2[]="China";
printf("%s", strcat(str1,str2));
```

要足够大

输出：PeopleChina

字符串处理函数strcpy

4. strcpy和strncpy函数-字符串复制

- strcpy一般形式为：

  strcpy(字符数组1,字符串2)

- 作用是将字符串2复制到字符数组1中去

```
char str1[10],str2[]="China";
strcpy(str1,str2);
```

要足够大

str1	C	h	i	n	a	\0	\0	\0	\0	\0

字符串处理函数strcpy

4. strcpy和strncpy函数-字符串复制

- strcpy一般形式为：

  strcpy(字符数组1,字符串2)

- 作用是将字符串2复制到字符数组1中去

```
char str1[10],str2[]="China";
strcpy(str1,str2);
```

必须是数组名形式

字符串处理函数strcpy

4. strcpy和strncpy函数-字符串复制

- strcpy一般形式为：

  strcpy(字符数组1,字符串2)

- 作用是将字符串2复制到字符数组1中去

```
char str1[10],str2[]="China";
strcpy(str1,str2);
```

数组名或字符串常量

字符串处理函数strcpy

4. strcpy和strncpy函数-字符串复制

- strcpy一般形式为：

  strcpy(字符数组1,字符串2)

- 作用是将字符串2复制到字符数组1中去

```
char str1[10],str2[]="China";
strcpy(str1,str2); 相当于
strcpy(str1,"China");
```

字符串处理函数strcpy

4. strcpy和strncpy函数-字符串复制

```
char str1[10],str2[]="China";
str1="China";          错误
str1=str2;             错误
strcpy(str1, str2);    正确
```

字符串处理函数strncpy

4. strcpy和strncpy函数-字符串复制

- 可以用strncpy函数将字符串2中前面n个字符复制到字符数组1中去
- strncpy(str1，str2，2);
  - 作用是将str2中最前面2个字符复制到str1中，取代str1中原有的最前面2个字符

strncpy注意事项

- strncpy有三个参数：（1）目标字符数组名（2）源字符串（3）一个整数n

  - ◆如果 目标字符数组长度>=参数n>源字符串长度，那么全部源字符串就会被赋值到目标字符数组中（包括'\0'）；
  - ◆如果 参数n<源字符串长度，则在源字符串中按指定长度n截取复制到目标字符数组中（不包括'\0'）；
  - ◆如果 参数n>目标字符数组长度，则发生语法错误

字符串处理函数strcmp

5. strcmp函数----字符串比较函数

- 其一般形式为

  strcmp(字符串1，字符串2)

- 作用是比较字符串1和字符串2

```
strcmp(str1,str2);
strcmp("China", "Korea");
strcmp(str1,"Beijing");
```

字符串处理函数strcmp

5. strcmp函数----字符串比较函数

- 字符串比较的规则是：将两个字符串自左至右逐个字符相比，直到出现不同的字符或遇到'\0'为止
  - ◆如全部字符相同，认为两个字符串相等
  - ◆若出现不相同的字符，则以第一对不相同的字符的比较结果为准

字符串处理函数strcmp

5. strcmp函数----字符串比较函数

"A"<"B"　　"a">"A"

"computer">"compare"

"these">"that"　"1A">"$20"

"CHINA">"CANADA"

"DOG"<"cat"

"Xiamen">"XIAMEN"

字符串处理函数strcmp

5. strcmp函数----字符串比较函数

- 比较的结果由函数值带回
  - ◆如果 字符串1=字符串2，则函数值为0
  - ◆如果 字符串1>字符串2，则函数值为一个正整数（即>0）
  - ◆如果 字符串1<字符串2，则函数值为一个负整数（即<0）

字符串处理函数strcmp

5. strcmp函数----字符串比较函数

```
if (str1>str2)
    printf("yes");     错误
if (strcmp(str1,str2)>0)
    printf("yes");     正确
```

字符串处理函数strlen

6. strlen函数----测字符串长度的函数

- 其一般形式为：

  strlen（字符数组）

- 它是测试字符串长度的函数
- 函数的值为字符串中的实际长度

字符串处理函数strlen

6. strlen函数----测字符串长度的函数

```
char str[10]="China";
printf("%d",strlen(str));
```

- 输出结果是5
- 也可以直接测试字符串常量的长度

```
strlen("China");
```

字符串处理函数strlwr

7. strlwr函数----转换为小写的函数

- 其一般形式为

strlwr (字符串)

- 函数的作用是将字符串中大写字母换成小写字母

```
char a[] = "HELLO";
strlwr(a);    注意：参数不能是字符串常量
printf("%s\n", a); //输出hello
```

字符串处理函数strupr

8. strupr函数----转换为大写的函数

- 其一般形式为

strupr（字符串）

- 函数的作用是将字符串中小写字母换成大写字母

```
char a[] = “hello”;
strupr(a);   注意：参数不能是字符串常量
printf(“%s\n”, a); //输出HELLO
```

字符数组应用举例：统计单词数

- 输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。
- 解题思路第一步：怎样统计有多少个单词
  - 方法：从第1个字符开始逐个字符进行检查，判断此字符是否是新单词的开头，如果是，就使变量num的值加1，最后得到的num的值就是单词总数

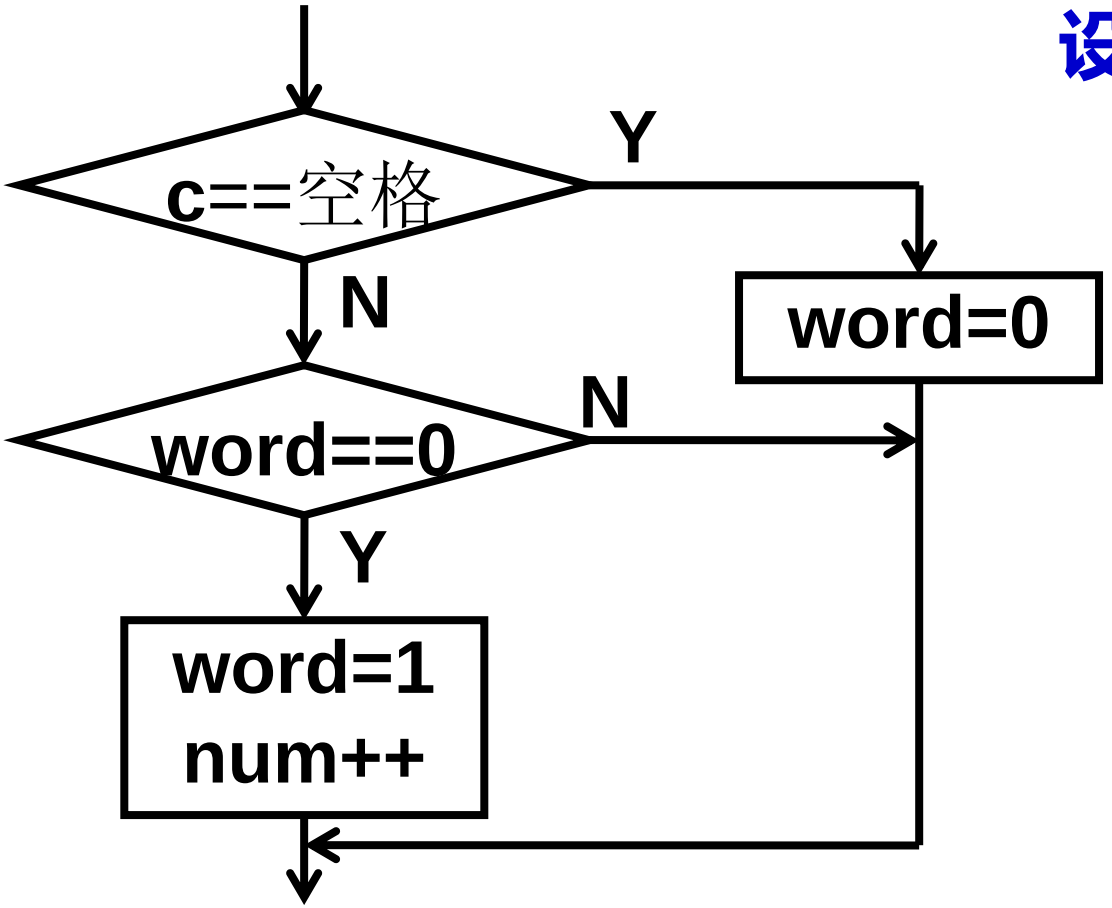
字符数组应用举例：统计单词数

- 输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。
- 解题思路第二步：怎样判断是否新单词的开头
  - 判断是否出现新单词，可以由是否有空格出现来决定(连续的若干个空格作为出现一次空格；一行开头的空格不统计在内)。如果测出某一个字符为非空格，而它的前面的字符是空格，则表示“新的单词开始了”，此时使num累加1；如果当前字符为非空格而其前面的字符也是非空格，则num不应再累加1

字符数组应用举例：统计单词数

- 输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。
- 解题思路第三步：**简化条件判断**
  - **可以用变量word作为判别当前是否开始了一个新单词的标志**，若word=0表示未出现新单词，如出现了新单词，就把word置成1
  - 前面一个字符是否空格可以从word的值看出来，若word等于0，则表示前一个字符是空格；如果word等于1，意味着前一个字符为非空格

字符数组应用举例：统计单词数



设C是当前检查的字符

```
if (c==' ')
{
    word=0;
}
else if (word==0)
{
    word=1;
    num++;
}
```

字符数组应用举例：统计单词数

当前字符	I		a	m		a		b	o	y	.
是否空格	否	是	否	否	是	否	是	否	否	否	否
word原值	0	1	0	1	1	0	1	0	1	1	1
新单词开始否	是	否	是	否	否	是	否	是	否	否	否
word新值	1	0	1	1	0	1	0	1	1	1	1
num值	1	1	2	2	2	3	3	4	4	4	4

```
if (c==' ')
{
    word=0;
}
else if (word==0)
{
    word=1;
    num++;
}
```

字符数组应用举例：统计单词数

```
……
1. char string[81],c;
2. int i,num=0,word=0;
3. gets(string);
4. for (i=0;(c=string[i])!='\0';i++)
5. {
6.     if (c==' ')
7.         word=0;
8.     else if (word==0)
9.     {
10.         word=1;
11.           num++;
12.    }
13.}
14.printf("%d words\n",num);
……
```

一定要设初始值

相当于
c=string[i];
c!='\0'

```
I am a boy.
4 words
```

作业 2017/11/15

- **按下列要求编写程序，提交手写源代码**

1. 有n盏灯，编号为1~n。第1个人把所有灯打开，第2个人按下所有编号为2的倍数的开关（这些灯将被关掉），第3个人按下所有编号为3的倍数的开关（其中关掉的灯将被打开，开着的灯将被关闭），依此类推。一共有k个人，问最后由哪些灯开着？输入：n和k，输出开着的灯的编号。0<=k<=n<=1000。样例输入：7 3，样例输出：1 5 6 7
2. 输入一个3x3的整数矩阵，把它左转90度后输出。例如：

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

- **上机练习（不用交）：本讲义例程，教材第六章2～8**