11.4-字符型数组(一)

字符型数组：char str[10];

1.字符型数组和字符串——’/0’字符

字符’\0’称为字符串结束符或串尾符，如果一维字符数组包含字符’\0’，则该字符之前的数组元素构成字符串。

字符型数组不一定都是字符串，因为字符型数组中并不是必须存在'\0'字符

如果字符数组中存在'\0' ，则'\0'及其前面的字符一起构成字符串， '\0'后的字符不属于该字符串。

在C++的标准函数库中字符串类，后面的课程再进行详细介绍。

2.声明和初始化：

char a[10]={'H','e','l','l','o'}; //没有’/0'，不是字符串

char b[]={'H','e','l','l','o'}; //数组大小为5，不是字符串

char c[10]={'H','e','l','l','o','/0'}; //是字符串

char d[10];

cin>>d;

char e[10]="Hello"; //是字符串，e[5]=’/0’

char f[]="Hello"; //是字符串，大小为6

眼花缭乱的0

①    ‘0’  代表    字符0  ，对应ASCII码值为48

②    '\0'    代表    空字符(转义字符)【输出界面显示为空】， 对应ASCII码值为   0x00(也就是十进制 0)， 用作字符串结束符

③     0    代表     数字0

④     “0”  代表    一个字符串，  字符串中含有 2个字符，分别是 '0' 和  '\0'

代码示例：

char ch1='\0'; //空字符

cout<<ch1<<"1"<<endl; //输出一个空字符(1是为了彰显空字符)

cout<<(int)ch1<<endl; //ASCII码为0

char ch2='0'; //字符'0'

cout<<ch2<<endl; //输出一个字符'0'

cout<<(int)ch2<<endl; //ASCII码为48

char ch3=0; //将ASCII码为a的字符'\0'赋值给ch3

//用数值给某个字符变量赋值时，相当赋与该数字相同码值所对应的字符

cout<<ch3<<"1"<<endl; //输出一个空字符(1是为a了彰显空字符)

cout<<(int)ch3<<endl; //ASCII码为0

//char ch4="0";

char ch4[]="0"; //"0"是一个字符串，不能赋值给单独的char型变量，应该用数组、指针等

cout<<sizeof(ch4)<<endl; //占了2个字节

cout<<ch4[0]<<" "<<(int)ch4[0]<<endl; //字符'0' ASCII码48

cout<<ch4[1]<<" "<<(int)ch4[1]<<endl; //字符'\0' ASCII码0

3.输入与输出：

 C++输入过程中，是把输入加载到缓冲区中，然后对缓冲区中的字符进行读取。cin，cin.get(),cin.getline()三个函数虽然都能进行数据读取，但是它们对缓冲区内数据的处理方法是不同的（如遇到[space],[enter]的处理方法)。这里将简单介绍cin,cin.get(),cin.getline()的基本用法，从根本上解释它们对缓冲区数据的处理方法。

（1）cin的基本用法

    结束条件：[enter],[space],[tab]

处理方法：cin遇到缓冲区中的[enter],[space],[tab]会结束当前输入，并舍弃[enter],[space],[tab]，继续下一项输入，当有连续[space],[enter,[tab]会全部舍弃。

输入：1234[enter]5678[enter]

输出：1234  【此处换行】   5678

输入：1234[enter][enter[enter]   ...   [enter]5678[enter]

输出：1234  【此处换行】 5678   （说明不管缓冲区中有几个enter，cin都会无视，直到不是enter开始继续读入）tab，space同理

综上：cin的输入很简单，整体的看输入，遇到[enter],[space],[tab]表示当前输入结束，下一个输入从非[enter],[sapce],[tab]开始，两输入间无线多个[space][]enter[tab]都是无效的

代码示例1：

char a[10]={'/0'},b[10]={'/0'};

//char a[10],b[10];

cin>>a;

cin>>b;

cout<<a<<"1"<<endl<<b<<"1"<<endl;

cout<<sizeof(a)<<" "<<sizeof(b)<<endl; //都是10，再次印证’\0’的确是一个字符

int i;

for(i=0;i<10;i++)

{

cout<<a[i]; //不初识化，输出会出现“烫烫烫”

}

cout<<endl;

for(i=0;i<10;i++)

{

cout<<b[i];

}

（2）.cin.get()基本用法

a.一参数

用法：a = cin.get()  或者  cin.get(a)

结束条件：输入字符足够后回车

说明：这个是单字符的输入，用途是输入一个字符，把它的ASCALL码存入到a中

处理方法：与cin不同，cin.get()在缓冲区遇到[enter]，[space]，[tab]不会作为舍弃，而是继续留在缓冲区中

代码示例2:

char a,b,c,d;

a=cin.get();

cin.get(b);

c=cin.get();

cin.get(d);

cout<<int(a)<<" "<<int(b)<<" "<<int(c)<<" "<<int(d)<<endl;

输入：[enter][enter][enter][enter]      输出：10,10,10,10

输入：abcd ef[enter]     输出：97,98,99,32   字符太多时取前几个，空格对应ASCII码为32

输入：[tab][enter][space]a[enter]     输出：9,10,32,97（分别为[tab]，[enter]，[space]，a的ASCALL码）不舍弃[enter],[tab],[space],统一看成字符

综上，一参数的cin.get()当输入字符足够后回车才能结束输入，把回车[enter],空格[space]也看成字符，在缓冲区中遇到不舍弃。当输入过多时，保留前几个。

（2）二参数

用法：cin.get(arrayname,size)

把字符输入到arrayname中，长度不超过size

注意：arrayname必须是char[]类型，即char数组类型，不可以为string类型；size是指输入长度，即允许输入的最大长度。

处理方法：假设把数据输入到 char a[arraylength],数组长度为arraylength。cin.get(a,size)---size,arraylength不同可以分为四类：

类1：输入串长<size，输入串长<arraylength，把串全部输入，后面补‘\0’

例1：char a[10]={'\0'};

cin.get(a,20);

cout<<a<<" "<<sizeof(a)<<endl;

输入：12345[enter]

输出：12345 10，此时数组内数据为‘12345‘\0’’

类2：输入串长<size，输入串长>arraylength，会自动扩张arrayname大小，使其能保存所有数据

输入：123456789abc[enter]

输出：123456789abc 10

可以发现，输入12个字符到a[10]中，系统自动扩充a[10]，此时实际数组长为13（‘123456789012'\0'’）。但当计算sizeof(a)时，还是10

虽然成功输出，但爆出了警告“栈溢出”（数组超限）

类3：输入串长>size，只截取size-1个字符，并在末尾补’\0’，放入数组中（若截取的串长>arraylength，数组可自动扩张）

输入：0123456789abcdefghijklmn

输出：0123456789abcdefghi 10

（3）三参数

用法：cin.get(arrayname,size,s)  把数据输入到arrayname字符数组中，当到达长度size时结束或者遇到字符s时结束

注释：a必须是字符数组，即char a[]类型，不可为string类型；size为最大的输入长度；s为控制，遇到s则当前输入结束且缓存区里的s将被舍弃

代码示例3：

char a[10]={'\0'};

cin.get(a,10,'.'); //再换成空格

cout<<a<<" 1"<<endl;

输入：abc.123

输出：abc    遇到  ’.‘ 输入结束

3.cin.getline()基本用法

cin.getline()的用法和cin.get()用法大致相同，没有一参数用法，只有二参数，三参数两类

cin.gerline(arrayname,size)

cin.getline(arrayname,size,s)用法与cin.get()效果大致相同，但也有些区别

（1）cin.getline(arrayname,size)与cin.get(arrayname,size)的区别

cin.get(arrayname,size)当遇到[enter]时会结束目前输入，他不会删除缓冲区中的[enter]

cin.getline(arrayname,size)当遇到[enter]时会结束当前输入，但是会删除缓冲区中的[enter]

char a[10]={'\0'};

char b;

cin.get(a,10);

cin.get(b);

cout<<a<<"#"<<endl<<(int)b;

输入：12345[enter]

输出：12345

10

说明：cin.get(a,10)把12345到a中，遇到[enter]结束，但是不把[enter]删除，则把[enter]输入到b中，（enter的ASCALL码为10）

char a[10]={'\0'};

char b;

cin.getline(a,10);

cin.get(b);

cout<<a<<"#"<<endl<<(int)b;

输入：12345[enter]a[enter]

输出：12345【换行】97

说明：cin.getline(a,10)吧12345输入到a中，遇到{enter}结束，并把缓冲区中[enter]删除，把接下来的a输入到b中

（2）cin.getline(arrayname,size,s)与cin.gei(arrayname,size,s)的区别

cin.get（arrayname,size,s）当遇到s时会结束输入，但不会删除缓冲区中的s

cin.getline(arrayname,size,s)当遇到s时会结束输入，并把s从缓冲区中删除

char a[10]={'\0'};

char b;

cin.get(a,10,',');

cin.get(b);

cout<<a<<"#"<<endl<<b;

输入：abc,0

输出：abc

，

说明：cin,get不会删除缓冲区的’，’

char a[10]={'\0'};

char b;

cin.getline(a,10,',');

cin.get(b);

cout<<a<<"#"<<endl<<b;

输入：abc,0

输出：abc

0

说明：cin.getline吧12345输入到a中，遇到’，‘结束，并删除缓冲区的’，‘，后面的字符0输入到b中

4.cin、cin.get、cin.getline的冲突

cout<<"请输入您的学号"<<endl;

int num;

cin>>num;

//cin.get();

cout<<"请输入您的姓名"<<endl;

char name[10]={'\0'};

cin.get(name,20);

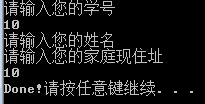
cout<<"请输入您的家庭现住址"<<endl;

char address[50]={'\0'};

cin.get(address,50);

cout<<num<<" "<<name<<" "<<address<<endl;

cout<<"Done!";



cout<<"请输入您的学号"<<endl;

int num;

cin>>num;

//cin.get();

cout<<"请输入您的姓名"<<endl;

char name[10]={'\0'};

cin.getline(name,20);

cout<<"请输入您的家庭现住址"<<endl;

char address[50]={'\0'};

cin.getline(address,50);

cout<<num<<" "<<name<<" "<<address<<endl;

cout<<"Done!";

