* 关于sizeof和strlen。Sizeof（）用于计算某类型或者某变量在内存中所占空间。比如整数分为short型，int型，long整型，分别占2,2/4,4个字节，int型具体占用几个字节和编译系统有关。我们输入字符串时通常用char数组：

Char s[10];

Scanf(“%s”,s);

输入的字符串可能是中英文混杂的，英文就是一个个字母，一个字母就是一个char，占一个字节。而一个汉字占用两个字节，这两个字节是连续的，每个字节最高位是1.所以可以依据这个判断一个字符串里面有多少汉字。

L=strlen(s); //这里用strlen，只是字符串长度（以字节为单位），不包括/0

For(num=i=0;i<l;i++)

{

If(s[i]<0) //有符号类型情况下，最高位1代表负数

Num++;

Printf(“共%d个汉字”,num/2); //这里要除以2，因为两个最高位为1的字节才是一个汉字

* //#include "stdafx.h"

#include <stdio.h>

#include <string>

#include <iostream>

using namespace std;

int main(void)

{

char a[] = { 'h', 'e', 'l', 'l', 'o' ,'\0'};

char b[] = "hello";//定义字符串

char c[] = { 'h', 'e', 'l', 'l', 'o' };

char \*p = b;

cout << a << endl;

cout << b<<endl;//输出的是hello

cout << c << endl;

char \*pchar = c;

//以下语句实现用指针输出hello。

for (size\_t i = 0; i < 5; i++)

{

cout << \*pchar;

pchar++;

}

cout << "\n" << endl;

cout << \*p;//指针指向首地址，所以输出为 h

string str = "hello world";

string \*p1 = &str; //注意必须加取地址运算符 &

cout<< str << "," << \*p1; //输出的是 hello world,hello world

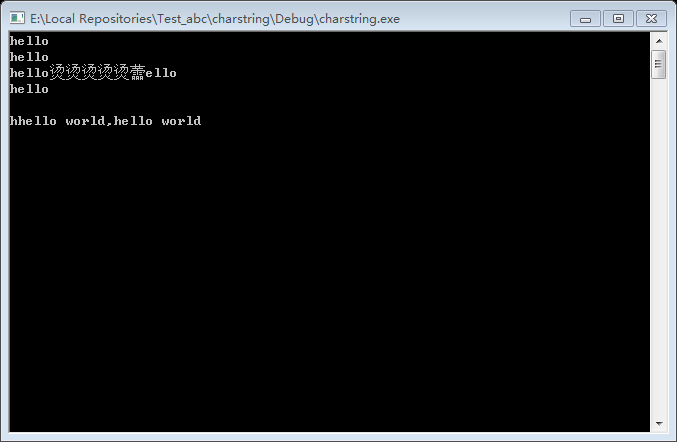
getchar();

return 1;

}Char数组可以存放字符串，而且数组名就是字符串的地址。Scanf和cout都是对地址进行操作，所以cout<<b结果就是字符串。

Char数组按字母存放时，要想输出字符串，要加字符串结束标识符\0，否则输出的是乱码。或者也可以利用指针一个个输出字母

String其实是一个类，而不是数据类型。String的指针指向string对象，而不是第一个字符。

二者之间是可以相互转换的。

* 说到类，其实类就是一个构造数据类型，即用已经定义好的数据类型重新构造的新的数据类型。除了类，还有数组类型，结构体类型，枚举类型，联合体类型。

C语言中的结构体类型和c++中的类类型基本相同，只不过默认的限定符不同，c语言默认public，c++默认private

* C++的麻烦之处就在于调用变量的方式。可以使用变量名，可以使用指针，还可以使用引用。程序实际是由函数构成的，包括多个子函数和一个主函数。而函数中参数的传递一般是按照值传递的，按照值的意思是会创建一个变量的副本，这个副本的值和参数的值相等。这样副本的值在函数中改变的话并不会改变原来的变量的值。
* 引用类型，在定义的时候在变量前面加一个&：

Int a;

Int &b=a;&c=a; //b,c是a的两个引用

引用变量是一个变量的别称，没有分配新的内存空间，这样改变变量的值，所有的引用变量的值都会改变。

* 说到函数与变量，还有一个概念那就是全局变量和局部变量，判断的标准就是变量的定义放在函数外面还是函数内部。但全局变量和局部变量的区别没有这么简单，全局变量分配的空间位于内存空间中的“堆”，堆的存取速度慢，空间大；局部变量的空间位于内存中的“栈”，栈存取速度快，空间小。而且全局变量定义时有默认的初始值，比如整型全局变量的默认初始值是0.
* 全局变量在所有函数中都可以访问，所以如果参数传递是引用的话，会在函数中造成改变，影响其他函数，所以不经常使用全局变量。而在堆中有空间大的优势，所以当我们要定义一个大的数组时，可以用静态变量。
* 静态变量以static限定符修饰。即便定义在函数体内部，程序跳出时依然存在，而且保持原来的值不变。静态变量的一个特点是只定义一次，只在函数首次调用时被定义，所以可以利用这个特点记录函数被调用的次数：在函数体中定义static int count=0;

Count++; 静态变量也默认初始化为0

* 数组的初始化。定义的时候如何没有同时对数组中任何一个元素赋初值，那么数组中每一个元素的值都是不确定的。注意这里的同时和任何。同时强调的是只能从头开始连续赋值，任何强调的是其余未赋值的会初始化为0.

#include<stdio.h>

#include<iostream>

using namespace std;

int main()

{

int i, j;

cout << "Case:1" << endl;

for (i = 0; i < 4; i++)

{

for (j = 0; j <= i; j++)//小于等于怎么写？

{

cout << "\*";

}

cout << endl;

}

cout << "\n";

cout << "Case:2" << endl;

for (i = 0; i < 4; i++)

{

for (j = 0; j < 4; j++)

{

if (j >= i)

cout << "\*";

else

cout << " ";

}

cout << endl;

}

cout << "\n";

cout << "Case:3" << endl;

for (i = 0; i < 4; i++)

{

for (j = 0; j <=3+i; j++)

{

if (j >=3- i)

cout << "\*";

else

cout << " ";

}

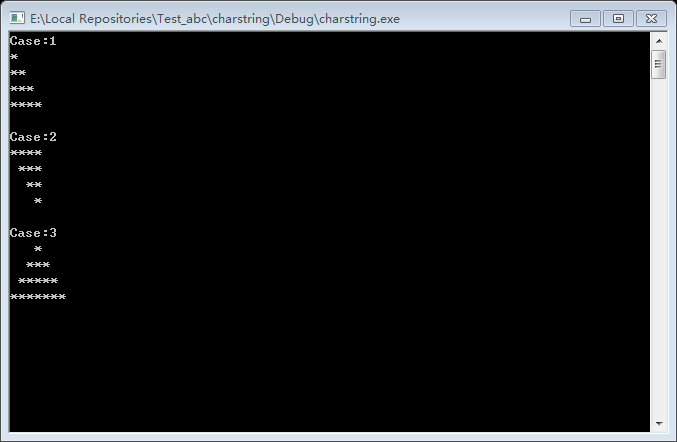
cout << endl;

}

getchar();

return 1;

}



* 上机考试总会有测试用例，测试用例都存放在文件中，这就需要从文件中读取数据。

While(scanf(“%d%d”,&a,&b)!=EOF)

{

}

EOF实际是一个宏定义，等于-1，意思是End Of File。而scanf读到文件尾也会返回-1.这时候才跳出循环。Scanf读取数据成功时，会返回成功读取的数据的个数

* 在循环体中要想读入数据显示出来，有格式化输入输出函数scanf("<格式化字符串>", <地址表>);printf("<格式化字符串>", <参量表>);还有非格式化输入输出函数gets(字符数组名或指针);puts()。后者占用内存更小，且可以读入空格，直到回车终止。Scanf遇到空格会停止，并在后面加终止符\0。puts（）在输出字符串时会将’\0’自动转换成’\n’进行输出，也就是说，puts方法输出完字符串后会自动换行。
* #define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

int main()

{

char a[10], b[10];

char c, d;

scanf("%s", a);

printf("a中保存的字符串为：%s\n", a);

c = getchar();

printf("c中保存的字符为：%c ", c);

gets(b);

printf("b中保存的字符串为：%s\n", b);

d = getchar();

printf("d中保存的字符为：%c", d);

getchar();

return 0;

}

Scanf遇到空格、Tab、回车时自动加\0停止输入，但之后的空格和回车仍然在缓冲区，会读入到之后的getchar中。为了避免这个问题，可以在两次输入之间插入一个getchar（）把这个回车符吃掉。或者在scanf的时候多写一个%\*c，读取最后一个回车符，但是不把它赋予任何变量。

gets：可接受回车键之前输入的所有字符，并用’\0’替代 ‘\n’.回车键不会留在输入缓冲区中

<https://blog.csdn.net/xingjiarong/article/details/47282817>

scanf 函数报错 error C4996: 'scanf'，因为vs认为c标准函数不安全。可以依照提示添加一个宏定义<https://blog.csdn.net/jh0703/article/details/47820875>

* Gets时不需要考虑EOF

Char str[10];

While(gets(str))