**String**

string包含在头文件<string>中，string定义在命名空间std中，所以若想使用string就要加入以下代码。结尾自动添加\0。string字符串可以看做数组，可以使用下标访问单个字符。

#include <string>

using std::string

**定义及初始化**

string s1;

string s2 = s1;

string s3 = "qwer";//拷贝初始化，将右侧的值创建为临时变量，拷贝到左侧新创建的变量中

string s4("wqer123"); //直接初始化

string s5(2,'d'); //第二个参数如果包含多个字符，只会使得最后一个字符起作用

string s6=”qq\0EE”; // \0后的内容不会被赋值，可以使用下标法访问进行测试

**内置方法**

cout << s3.empty() << endl; //如果为空则返回true ，不为空返回false

cout << s3[0] << endl; //以下标方式访问字符，从零开始

**获取长度**

size和length遇到\0便结束，不会将\0记入

cout << s3.size() << endl; //返回字符个数

cout << s3.length() << endl; //返回字符串长度

cout << s3.capacity() << endl; //返回容器大小

**输入**

cin获取输入时自动忽略开头的空白，即空白符、制表符、换行符等，从第一个真正的字符开始读取，到下一个空白结束

举例

现在输入“ hello world”

cin>>s3

s3中实际存放的是”hello”

getline也用来输入，它可以将输入流完整地保存下来，getline与cin的不同在于getline可以保存空格，使用换行符结束

举例，现在输入“ hello world”

getline(cin,s5)

s3中实际存放的是“ hello world”

**字符串的拼接**

使用加号进行字符串拼接时，应注意加号两边至少一个为string变量，不能两边同时为 字面量。执行到string变量或字面量的第一个\0时，便会结束当前string变量或字面量 的赋值

s3+s4 //将两个string变量连接并返回,结果会自动去除第一个变量的\0

s3+”qwer” //左侧为string变量，右侧为字面量

”qwer”+“，” //两侧都为字面量，错误

s3+”qw\0er” //当执行到”qw\0er”中的\0时便会结束当前字面量的拼接

**基于范围for的字符串逐个迭代方法**

格式：

for(类型 变量名：字符串)

{

操作语句

}

这个结构可以将字符串的每个字符逐个赋值给变量，直到\0，不会将 \0后的内容 赋值过去。

变量类型多用auto，在对字符串使用时可以直接写成char类型。

这个结构还适用于其他类型的数组，使用时注意变量类型。

可以使用别名对其内容进行修改，如：for(auto &str:string) { }

字符判断控制函数

isainum(c) 当c为数字或字母时为真

isalpha(c) 当c为字母时为真

isdigit(c) 当c为数字时为真

iscntrl(c) 当c为控制字符时为真

islower(c) 当c为小写字母时为真

isupper(c) 当c为大写字母时为真

tolower(c) 当c为大写字母时返回对应小写字母，否则原样返回

toupper(c) 当c为小写字母时返回对应大写字母，否则原样返回

习题

1. 用户输入两个字符串，将两个字符串进行比对，将中第一个不同字符的ascll码差值打印出来，不得打印负数
2. 用户输入一串包含大小写的字符，输出这个字符串，请将原来大写的字符变为小写，小写的字符变为大写，然后输出这个字符串
3. 用户输入一个字符串，然后再输入想要删除的字符，请将这个删除后的字符串输出