使用字符数组存放字符串，并在此基础上进行字符串运算，是C语言的方法，被C++保留了下来，使用这样的方法必须定义一个字符数组，但是字符数组是有一定大小的。在进行字符串连接或字符串复制时，如果未能准确计算字符串和字符数组的长度，就会发生将一部分字符放在字符数组范围之外，从而可能破坏系统的正常工作状态。因此使用字符数组来存放字符串并不是最理想和安全的方法。

C++可以使用一种更方便、安全的方法存放字符串，即新的数据类型"字符串类型"(string)，在使用方法上和大多数基本数据类型一样，使用一个标识符代表字符序列。

实际上string并不是C++语言本身具有的数据类型，它是在C++标准库中声明的一个字符串类，使用这种类可以定义对象（作类使用必须包含头文件<string>）。

1、字符串变量的定义和初始化

和其他类型变量一样，字符串变量必须先定义后使用，定义了字符串变量之后，可以用复制语句对其赋予一个字符串常量。

string str = "Hello world!";

或

string str;

str = "Hello world!"; //这是字符数组不可以使用的方式

需要注意，使用string保存的字符串不再包含结束符'\0'

既可以用字符串常量给字符串变量赋值，也可以用一个字符串变量给另一个字符串变量赋值

如string\_2 = string\_1，不要求string\_2和string\_1长度相同，在定义字符串变量时不需指定长度，它的长度随其中的字符串长度而改变。

可以对字符串变量中某一字符进行操作

例如： string str\_1 = "Than";

str\_1[2] = 'e';

cout << str\_1 << endl; //输出的结果为Then

1. 字符串变量的运算

C++中字符串的运算可以不再使用字符串相关的函数，而是直接用简单的运算符

str\_1 = str\_2; //字符串复制（拷贝）

str\_1 += str\_2; //字符串拼接

str\_1 < str\_2; //字符串比较

1. 字符串数组

不仅可以用string定义字符串变量，也可以用string定义字符串数组

例如：string name[5]; //定义一个字符串数组，包含5个字符串元素

for(int i = 0;i < 5;i++){

getline(cin,name[i]);

}

for(int i = 0;i < 5;i++){

cout << name[i] << "\t" << name[i].length() <<endl;

}