**实验六 字符串处理**

**一、实验目的和要求**

输入一个字符串（少于80个字符），去掉重复的字符之后，按照字符的ASCII码值的从大到小输出。

**二、实验内容**

1. 实验准备
   1. 理论知识介绍
      1. 数据的输入：

要从用户依次读入int类型的变量a,n，可以使用如下语句:

cin >> a>>n;

* + 1. 数据的输出

要将变量的内容显示在显示器上，可以使用cout和流插入运算符。Cout还可以输出表达式的执行结果，比如:

cout << a << ‘+’ << b << ‘=’ << a+b << endl;

* + 1. 算术表达式

C++中，算术运算里有+（加法），-（减法），/（除法），%（取模），但是没有取平方算数运算。

* + 1. 数据类型 （整型）：

C++中一个整型变量可以储存一个整数，可以直接通过cin和cout输入和输出。

* + 1. 关系表达式

关系表达式用于比较两个值的大小。C++提供了6个关系运算符: <, <=, >, >=, ==, !=。关系运算符左结合。

* + 1. for语句

**for (i = 0; i < n; ++i)**

**需要重复执行的语句**

* + 1. 字符串的定义

字符串的定义的语句可以是这样的：

**char ch[size];**

* + 1. 字符串的输入/输出

有三种方法：

1. 逐个字符的输入/输出，这种做法的普通的数组的操作一样；
2. 将整个字符串一次性地用对象cin和cout的>>和<<操作完成输入或输出；
3. 通过cin的成员函数getline输入。
4. 实验项目
   1. 分析
      1. C++中的字符串和数组一样，长度必须是常量表达式，程序要在编译的时候确定字符串的长度。题中给出字符的长度不超过80，所以我们在定义字符串的时候将字符串的长度设置为80，用cin.getline的函数的时候传入参数80。
      2. 读取字符串之后，由于根据ASCII码值进行排序，所以要选择一个排序的方法，我们目前学习过两种排序的方法，其中冒泡排序的效率比较高，所以我们采取冒泡排序的方法。但是采取冒泡排序的方法我们要事先知道真个字符串的长度，所以我们还要利用一个循环来获得字符串的长度。
      3. 进行排序之后，我们还要将重复的字符去掉，所以要设置另一个循环来去掉重复的字符。
   2. 方案
      1. 利用cin.getline来将用户的输入读取到一个字符串之中
      2. 利用for循环来获得字符串的长度
      3. 利用冒泡排序来将字符串中的字符以ASCII码的值来进行排序
      4. 利用for循环除去字符串中的重复的字符
      5. 利用for循环打印字符串中的字符
   3. 源程序

#include <iostream>

using namespace std;

const int maxsize = 80;

int main() {

int i,j,size;

char tmp;

bool flag;

char ch[maxsize];

cout << "Please input a string:";

cin.getline(ch, maxsize);

for (i = 0; ch[i] != '\0';i++) //Get the size of the string

size = i+1;

for (i = 1; i <= size; i++) { //Rank the string by ASCII

flag = false;

for (j = 1; j <= size - i; j++) {

if (ch[j - 1] < ch[j]) {

tmp = ch[j - 1];

ch[j - 1] = ch[j];

ch[j] = tmp;

}

}

if (flag)

break;

}

for (i = 1; i <= size; i++) { //Eliminate the repeated char

while (ch[i - 1] == ch[i]) {

for (j = i; j <= size; j++) {

ch[j - 1] = ch[j];

}

size--;

}

}

for (i = 1; i <= size; i++) { //Print out the string

cout << ch[i - 1];

}

return 0;

}

* 1. 测试数据
  2. ya7bb2tizx4m55n9q2
  3. ababab121212 cdcdcd343434
  4. Please input a string
  5. 120 years of SJTU!
  6. 运行结果

1. zyxtqnmiba97542
2. dcba4321
3. utsrpnligeaP
4. ysrofeaUTSJ210!

（注意ii, iii, iv这三个运行结果后面都有一个空格）

* 1. 异常现象

最初的程序在输入一个字符串之后，会不断运行，进入一个死循环，并不会输出结果。

* 1. 分析

最初的去掉重复的循环的语句是这样的，并没有红色的这一行。

for (i = 1; i <= size; i++) { //Eliminate the repeated char

while (ch[i - 1] == ch[i]) {

for (j = i; j <= size; j++) {

ch[j - 1] = ch[j];

}

size--;

}

}

经过调试分析，发现在这个循环体中会出现死循环的情况，这是因为利用这种方法去除多余的字符会让“aabbccdd”的字符串变成类似“abcddddd”的情况，如果不对字符串的长度进行动态的修改，当i=4时，while循环成立，当执行了一次while循环之后，字符串还是“abcddddd”,所以最后会出现while循环一直成立的情况，所以会出现死循环的情况。

* 1. 总结

在处理字符串的时候，如果涉及到字符串的长度的改变的话，要实时修改字符串的长度，否则会出现不可预料的错误。

**三、实验小结**

1. 完成情况

在添加了size--；这条语句之后输出了正确的结果。

1. 重点

字符串的初始化，输入/输出，以及字符串的数据的处理。

1. 难点

处理字符串的时候循环控制行中的循环变量的初始化和循环条件的选择，以及字符串的长度的动态处理。

1. 解决方法

将循环变量从1开始计数，较符合习惯。并将字符串的长度放置到一个变量size里面，每重复一次while循环就将size的值减少一次。

1. 有待改进之处

可以讲字符串的排序和字符串的多余字符的删除合并到一个循环体中。

1. 收获、体会

字符串的长度是一个常量表达式，但是我们可以将我们想要的字符串的长度放置在一个变量值中，将字符串中的其他的值忽略，只输出这个字符串中的一段特定长度的字符。