**暨南大学本科实验报告专用纸**

课程名称 C++程序设计 成绩评定

实验项目名称 标准库类型string的使用 指导教师 王勇杰

实验项目编号 2 实验项目类型 编程型 实验地点 C105

学生姓名 李炜鹏 学号 2017052544

学院 智能科学与工程学院 系 计算机 专业 信息安全

实验时间2019年12月10日 午～12月13日 午 温度 ℃湿度

1. **实验目的**
2. 理解标准库类型string的概念以及基本用法
3. 理解索引类型size\_type
4. 理解迭代器iterator
5. 熟悉for语句和while语句对string类的遍历，对比索引遍历和迭代器遍历的用法
6. 了解并熟悉C++提供的字符判断函数和字符转换函数isupper、toupper、islower、tolower、isalpha、isspace、ispunct等
7. 实验要求：
   1. 阅读课本p83的“使用范围for语句改变字符串中的字符”内容，完成课本p86的练习3.6的编程要求。
   2. 试分别调试运行实验二中的三个源代码 (见FTP服务器的“C++程序设计实验”文件夹。
8. 实验报告要求：
   1. 实验目的
   2. 实验原理
   3. 道出string的知识要点
   4. 注释必要的代码
   5. 实验结果的截图
   6. 实验评估
9. **实验原理**
10. string类：

string类作为一个类，不同于C语言，是C++新增加的面向对象的特性，string类里面包含成员和成员函数，其中size\_type、iterator、begin()、end()、size()较为常用，我们还可通过索引（index）或者迭代器（iterator）对string遍历，string类还有多种构造函数，我们可以使用多种方式对string初始化，string类还重载了多种运算符，比较常用的是（+、+=、=、==）

1. 索引类型size\_type是string类里面的一个成员，它作为无符号整型，用于作为string类的下标/索引
2. 迭代器iterator作为string类的字符指针，用于指向string类的其中一个字符，可以使用\*来获取当前字符的值
3. **主要仪器设备**

**仪器：**计算机

**实验环境：**Visual Studio 2017或Dev-C++

1. **源程序**
2. **实验要求一（P86练习3.6）：**

#include <iostream>

#include <string>

using namespace std;

int main()

{

cout << "Please enter a string: " << endl;

string str;

cin >> str;

for (decltype(str.size()) index = 0; index != str.size() && !isspace(str[index]); ++index)

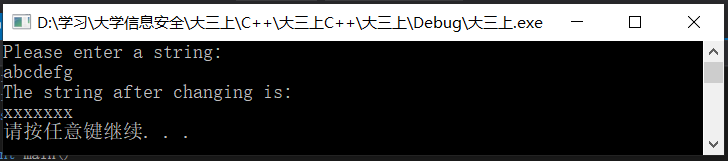
str[index] = 'x';

cout << "The string after changing is: " << endl << str << endl;

system("pause");

return 0;

}



1. **实验要求二（调试三个源代码）：**

**cctype.cpp**

#include <string>

using std::string;

#include <cctype>

using std::isupper; using std::toupper;

using std::islower; using std::tolower;

using std::isalpha; using std::isspace;

#include <iostream>

using std::cout; using std::endl;

int main()

{

string s("Hello World!!!");

//用于计算输入的字符串的标点符号的个数

string::size\_type punct\_cnt = 0;//punct\_cnt为string类的size\_type类型

for (string::size\_type c = 0; c != s.size(); ++c)

if (ispunct(s[c])) //判断当前字符是否标点符号

++punct\_cnt; //punct\_cnt累计加一

cout << punct\_cnt

<< " punctuation characters in " << s << endl;

//使用for语句遍历string(直到string的结尾，不忽略空格符)

string orig = s;

for (string::size\_type c = 0; c != s.size(); ++c)

s[c] = toupper(s[c]);//将字符串s转换成大写字母

cout << s << endl;

//使用while语句遍历string（遇到字符串结尾标志或者遇到空格符则结束循环）

s = orig; //重置s为原来的字符串orig

string::size\_type index = 0;

while (index != s.size() && !isspace(s[index]))

{

s[index] = toupper(s[index]);//将字符串s转换成大写字母

++index;

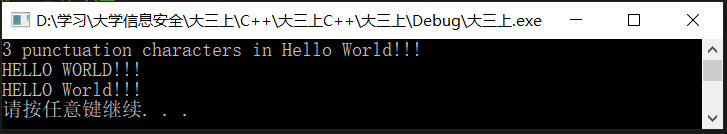
}

cout << s << endl;

system("pause");

return 0;

}

****

**char\_repl.cpp**

#include <string>

using std::string;

#include <iostream>

using std::cout; using std::endl;

int main()

{

string str("some string"), orig = str;

if (!str.empty()) //如果使用字符串的索引功能，必须首先确保字符串不空

cout << str[0] << endl;//使用下标（索引）输出字符串中的第一个字符

if (!str.empty())

str[0] = toupper(str[0]);//将字符串中第一个字符转为大写

cout << str << endl;

str = orig; //重置字符串s为初始字符串orig

//使用string类的迭代器遍历string类字符串

if (str.begin() != str.end())//通过判定字符串头≠字符串尾来确保字符串不空

{

string::iterator it = str.begin(); //string类迭代器it指向str的第一个字符

\*it = toupper(\*it);//通过\*运算符获取当前字符的值，并且转为大写

}

cout << str << endl;

str = orig;//重置字符串s为初始字符串orig

// 四种将字符串转为大写的方法:

// 1. 使用for语句遍历string（使用下标/索引）

for (string::size\_type index = 0; index != str.size() && !isspace(str[index]); ++index)

str[index] = toupper(str[index]); // 将当前字符转为大写

cout << str << endl;

str = orig;//重置字符串s为初始字符串orig

// 2. 使用for语句遍历string（使用迭代器iterator）

for (string::iterator it = str.begin(); it != str.end() && !isspace(\*it); ++it)

\*it = toupper(\*it); // 将当前字符转为大写

cout << str << endl;

str = orig;//重置字符串s为初始字符串orig

// 3. 使用while语句遍历string（使用下标/索引）

string::size\_type index = 0; // 定义下标/索引用于遍历string

while (index != str.size() && !isspace(str[index]))

{

str[index] = toupper(str[index]); // 将当前字符转为大写

++index; // 下标+1

}

cout << str << endl;

// 4. 使用while语句遍历string（使用迭代器iterator）

string::iterator beg = str.begin();

while (beg != str.end() && !isspace(\*beg))

{

\*beg = toupper(\*beg);

++beg;

}

cout << str << endl;

str = orig;//重置字符串s为初始字符串orig

// 使用for语句遍历字符串（使用迭代器iterator）

// 输出str字符串，每个字符之间用换行隔开

for (string::const\_iterator c = str.begin(); c != str.end(); ++c)

cout << \*c << endl; // 输出当前字符，并且使用换行符隔开

// 修改字符串

for (string::iterator c = str.begin(); c != str.end(); ++c)

\*c = '\*'; // 使用‘\*’来替代当前字符

cout << str << endl;

str = orig;//重置字符串s为初始字符串orig

// 使用for语句遍历字符串（使用下标/索引）

// 输出str字符串，每个字符之间用换行隔开

for (string::size\_type ix = 0; ix != str.size(); ++ix)

cout << str[ix] << endl; // 输出当前字符，并且使用换行符隔开

// 修改字符串

for (string::size\_type ix = 0; ix != str.size(); ++ix)

str[ix] = '\*'; // 使用‘\*’来替代当前字符

cout << str << endl;

str = orig;//重置字符串s为初始字符串orig

// 使用for语句遍历字符串（使用迭代器iterator）

// 输出str字符串，每个字符之间用换行隔开

for (string::const\_iterator beg = str.begin(); beg != str.end(); ++beg)

cout << \*beg << endl; // 输出当前字符，并且使用换行符隔开

// 修改字符串

for (string::iterator beg = str.begin(); beg != str.end(); ++beg)

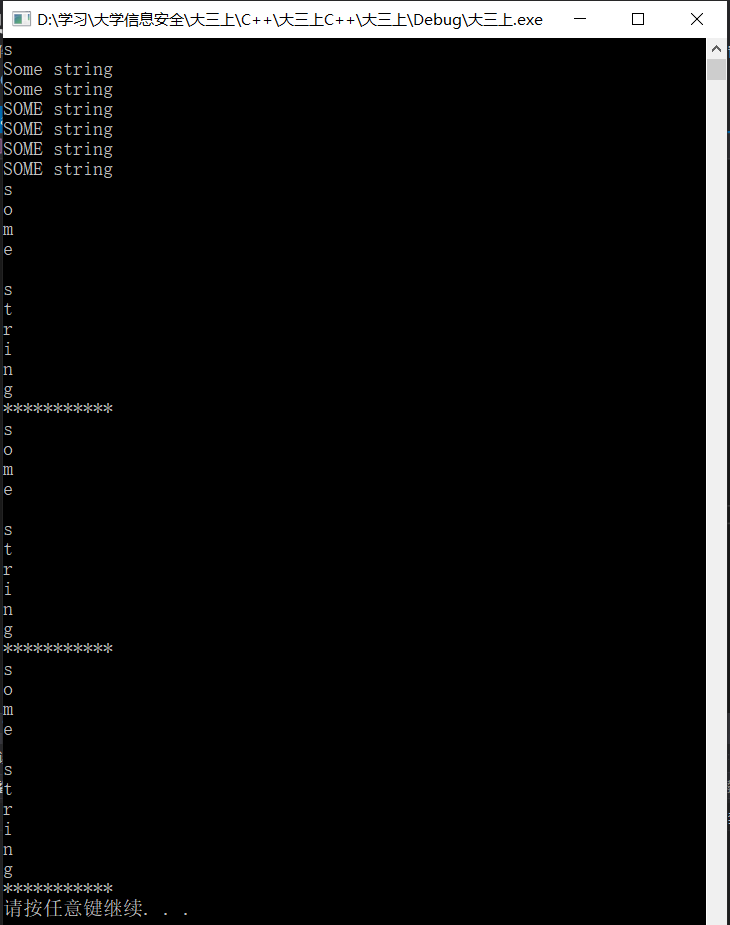
\*beg = '\*'; // 使用‘\*’来替代当前字符

cout << str << endl;

system("pause");

return 0;

}

****

上述代码实现的功能分别是：

输出首字母、将首字母转为大写并且输出整个字符串（两种方法：使用if (!str.empty())、使用if (str.begin() != str.end())）、将字符串第一个单词转为大写并且输出整个字符串（四种方法：使用for语句和索引、使用for语句和迭代器、使用while语句和索引、使用while语句和迭代器）、使用换行符隔开输出每一个字符并且将字符串使用\*修改后输出（两种方法，这里重复使用了其中一种方法，于是有三次输出）

**punctRemove.cpp**

#include <iostream>

#include <string>

#include <cctype>

using namespace std;

int main()

{

string s, result;

cout << "Please enter a string,it's best to include some punctuation marks. " << endl;

getline(cin, s);//接收cin的输入，输入到s字符串中，遇到空白符结束输入

//去掉string中的标点符号并输出新的字符串

for (decltype(s.size()) i = 0; i < s.size(); i++)

{

if (!ispunct(s[i]))

result += s[i];//string类里面重载了+=运算符，于是这里可以这样给result赋值

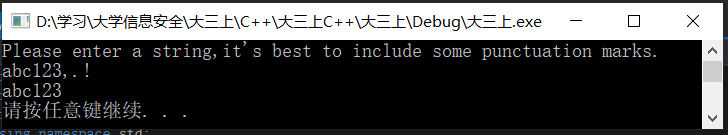
}

cout << result << endl;

system("pause");

return 0;

}

****

1. **实验评估**

通过本实验收获良多，我理解到了string类的概念，上机实现并熟悉了string类的用法，包括其数据成员（size\_type、iterator）、成员函数（begin()、end()、size()），熟悉了string类字符串的遍历（for循环、while循环、索引遍历、迭代器遍历），认识了字符判断函数、字符转换函数。