 实 验 报 告

课程名称： 数据结构（C++版）

题 目： 13.1串操作的实现

专业班级： 18空间

姓 名： 邓珊珊

学 号： 1812001107

指导老师： 袁莹

实验时间： 2019.11.21

第一次实验报告

班级： 18空间 姓名： 邓珊珊 学号：1812001107

1. 实验目的及原理

（选择公式的原理，及文字说明，所选的公式需要输入什么，计算出什么，如何计算，目的：通过实际问题学会用函数编程实现）

1. 实验目的
2. 掌握串的顺序存储结构；
3. 验证顺序串及其基本操作的实现；
4. 掌握串的操作特点。
5. 实验内容
6. 定义一个包含串的求长度、拼接、比较大小等基本操作的头文件函数原型；
7. 实现串的求长度、拼接、比较大小等基本操作。
8. 实验提示

在C/C++语言中，采用顺序存储结构来存储字符串，并且字符串的尾部存储一个特殊字符’\0’(ASCII码为0）作为串的终结符。字符串的操作本质上属于数组的操作，需要注意如下两点：

1. 由于没有存储串的长度，因此，通常以终结符作为循环条件；
2. 可以设置一个指针指向存储字符串的起始地址，通过指针实现对字符串中特定字符的操作。
3. 实验内容

（代码）

头文件：

在VC++编程程序环境下新建一个工程“串操作的实现”，在该工程中新建一个头文件str.h，该头文件包括与串的基本操作对应的函数声明，范例程序如下：

#ifndef Str.H //避免重复包含str.h 头文件

#define Str\_H

int strlen(char \*s); //函数原型，求串的长度

char \* strcat(char \*s1,char \*s2); //函数原型，将s2拼接到s1的后面

int strcmp(char \* s1,char \*s2); //函数原型，比较字符串s1和s2的大小

#endif

str. cpp

#include "str.h"

int strlen(char \* s)

{

char \* p = s;

int len = 0;

while(\* p != '\0')

{

len++;

p++;

}

return len;

}

char \* strcat(char \* s1, char \* s2)

{

char \* p = s1, \* q = s2;

while(\* p != '\0')

p++;

while(\* q !='\0')

{

\* p = \* q;

p++ ; q++;

}

\* p = '\0';

return s1;

}

int strcmp(char \* s1,char \*s2)

{

char \* p = s1,\* q = s2;

while(\* p != '\0' && \* q != '\0')

{

if(\* p > \* q)

return 1;

else if(\* p < \* q)

return -1;

else{p++;q++;}

}

if(\* p =='\0' && \* q =='\0')

return 0;

if(\* p != '\0')

return 1;

if(\* q !='\0')

return -1;

else

return 0;

}

str\_main.cpp

#include <iostream>

#include "str.h"

using namespace std;

int main()

{

char ch[20]="I love", \* str =" China!";

cout<<strlen(ch)<<endl;

cout<<strlen(str)<<endl;

cout<<strcmp(ch,str)<<endl;

cout<<strcmp(str,ch)<<endl;

strcat(ch,str);

for(int i = 0;ch[i] != '\0';i++)

cout<<ch[i];

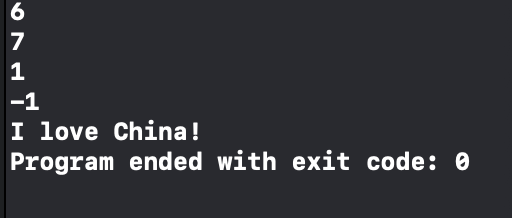
cout<<endl;

return 0;

}

1. 实验成果

（运行结果截图）



1. 实验心得

通过本次实验，使用for循环进行串的连接和串的复制，以及串的替换。在每一步的操作中，都需要指针的移动。如在串的连接中，当s2中的一个字符天添加到s1后时，s1、s2的指针都要后移一个位置，并且s1串长度加1，循环到串s2结束。