# 实验报告

专业班级：软件工程2301

学号：8209230111

姓名：郑怡

# 实验三：

题1：

#include"head.h"

//求最大公约数

int GCD(int m, int n)

{

int t;

t = m > n ? n : m;

while (m % t != 0 || n % t != 0)

{

t--;

}

cout << "最大公约数为" << t << endl;

return t;

}

//求最大公约数和最小公倍数

int GL(int& m, int& n)

{

int t;

t = m > n ? n : m;

while (m % t != 0 || n % t != 0)

{

t--;

}

cout << "最大公约数为" << t << endl;

int k;

k = m > n ? m : n;

while (k % m != 0 || k % n != 0)

{

k++;

}

cout << "最小公约数为" << k << endl;

return t, k;

}

int main()

{

int m, n;

cout << "输入自然数m,n " << endl;

cout << "m= ";

cin >> m;

cout << "输入自然数m,n" << endl;

cout << "n= ";

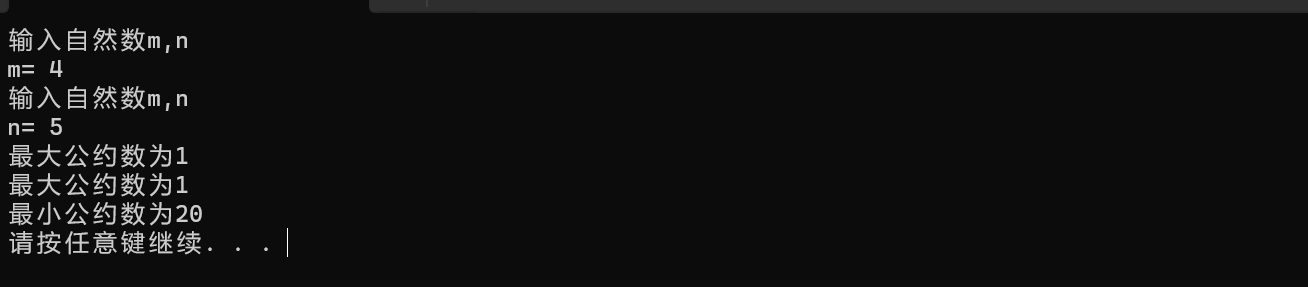
cin >> n;

GCD(m, n);

GL(m, n);

system("pause");

system("cls");

}

题2：

#include"head.h"

bool is\_prime(int num)

{

int t = 0;

if (num == 2)

{

return true;

}

else if (num > 2)

for (int i = 2; i < num; i++)

{

if (num % i == 0)

{

t++;

}

}

if (t == 0)

{

return true;

}

else if (t > 0)

return false;

}

int main()

{

//素数判断

int num;

cout << "输入一个整数" << endl;

cin >> num;

cout<<is\_prime(num)<<endl;

//找出前两百个素数

int prime = 0;

for (int i = 2; prime <= 200; i++)

{

if (is\_prime(i))

{

if (prime % 11 != 0)

cout << i << " ";

else

cout << endl;

prime++;

}

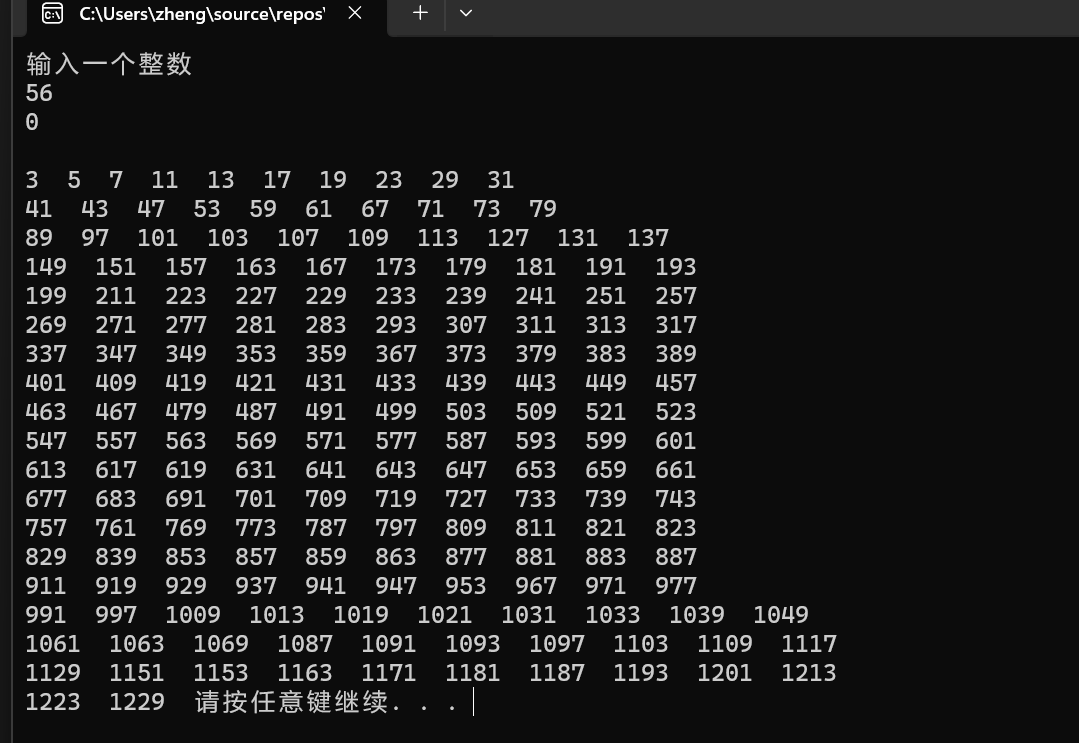
else

{ }

}

system("pause");

system("cls");

}

题3：

# mytemperature.h：

#pragma once

////摄氏温度转华氏温度函数声明

double celsius\_to\_fah(double cel);

//华氏温度转摄氏温度函数声明

double fahrenheit\_tocelsius(double fah);

# mytemperature.cpp：

#include"head.h"

//F = 9×C /5+32

double celsius\_to\_fah(double cel)

{

double fah;

fah = 9 \* cel / 5 + 32;

cout << "celsius" << '\t' << "fahrenheit" << '\t' << endl;

cout << cel << '\t' << fah << '\t' << endl;

return fah;

}

//C = 5×（F- 32）/9，

double fahrenheit\_tocelsius(double fah)

{

double cel;

cel = (fah - 32) \* 5 / 9;

cout << "fahrenheit" << '\t' << "celsius" << '\t' << endl;

cout << fah << '\t' << cel << '\t' << endl;

return cel;

}

题三代码：

#include"head.h"

#include"mytemperature.h";

int main()

{

double cel;

cin >> cel;

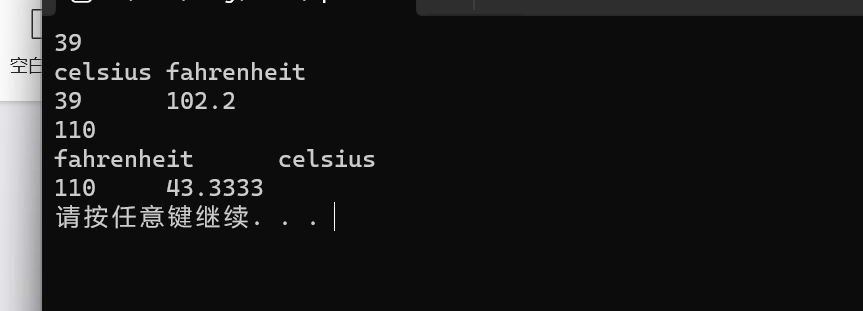
celsius\_to\_fah(cel);

double fah;

cin >> fah;

fahrenheit\_tocelsius(fah);

system("pause");

}

题5：

#include"head.h"

int main()

{

int t = 1;

int sum;

for (int i = 0; i < 10; )

{

i++;

sum = t;

cout << t << endl;

t = t \* 2 + 2;

}

cout << "猴子第一天摘的个数为：" << sum << endl;

}

# 实验四数组：

题1：

int main()

{

//输入10个数

int arr[10];

int sub = 10;

cout << "Enter ten numbers: ";

for (int i = 0; i < sub; i++)

{

cin >> arr[i] ;

}

for (int i = 0; i < sub - 1; i++)

{

for (int j = i + 1; j < sub; j++)

{

if (arr[i] == arr[j])

{

for (int k = j; k < sub-1; k++)

{

arr[k] = arr[k + 1];

}

sub--;

j--;

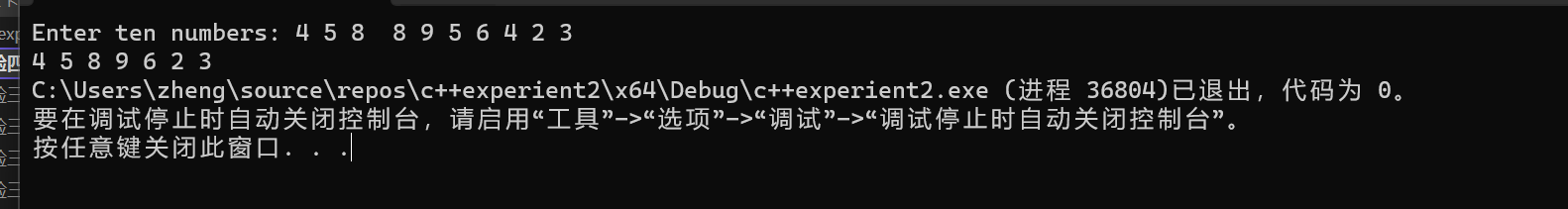
}

}

}

for (int i = 0; i < sub; i++)

cout << arr[i] << " ";

}

题2：

#include"head.h"

void list1(double list[])

{

bool changed = true;

do

{

changed = false;

for (int j = 0; j < 9; j++)

if (list[j] > list[j + 1])

{

swap(list[j], list[j + 1]);

changed = true;

}

} while (changed);

}

int main()

{

cout << "输入一个有十个元素的数组" << endl;

double list[10];

for (int i = 0; i < 10; i++)

{

cin >> list[i];

}

list1(list);

cout << "排序后" << endl;

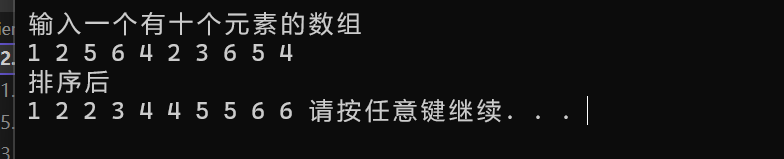
for (int j = 0; j< 10; j++)

{

cout << list[j] << " ";

}

system("pause");

}

题3：

#include"head.h"

int main()

{

bool L[100];

for (int i = 0; i < 100; i++)

{

L[i] = false;

}

int stu = 0;

for (stu; stu < 100; stu++)

{

for (int j = stu; j < 100; j=j+stu+1)

{

if (L[j] ==false)

{

L[j] = true;

}

else if(L[j]==true)

{

L[j] = false;

}

}

}

cout << "柜子开着的号码为";

for (int k = 0; k < 100; k++)

{

if (L[k])

{

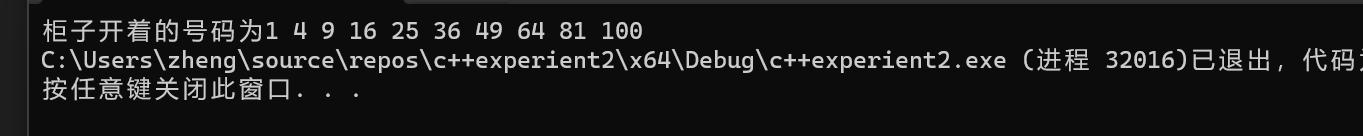
cout << k + 1 << " ";

}

else

{}

}

}

题4：

#include"head.h"

void merge(const int list1[], int size1, const int list2[], int size2, int list3[])

{

for (int i = 0; i < size1; i++)

{

list3[i] = list1[i];

}

for (int j = 0; j < size2; j++)

{

list3[size1 + j] = list2[j];

}

//冒泡排序

for (int k = 0; k < size1 + size2 - 1; k++)

{

for (int m = 0; m < size1 + size2 - k - 1; m++)

{

if (list3[m] > list3[m + 1])

{

int tem;

tem = list3[m];

list3[m] = list3[m + 1];

list3[m + 1] = tem;

}

}

}

cout << "The merged list is :";

for (int n = 0; n < size1 + size2; n++)

{

cout << list3[n] << " ";

}

}

int main()

{

int size1, size2;

cout << "Enter list1: ";

cin >> size1;

int \*list1;

list1 = new int[size1];

for (int i = 0; i < size1; i++)

{

cin >> list1[i];

}

cout << "Enter list2: ";

cin >> size2;

int\* list2;

list2 = new int[size2];

for (int i = 0; i < size2; i++)

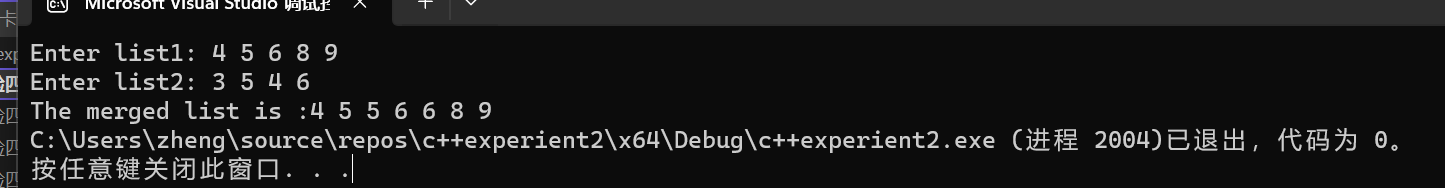
{

cin >> list2[i];

}

int\*list3 = new int[size1+size2];

merge(list1, size1, list2, size2,list3);

}

题5：

#include"head.h"

int indexOf(const char s1[], const char s2[])

{

//s1是s2的子串

int a, b;

a = strlen(s1);

b = strlen(s2);

if (a > b)

{

cout << "s1不是s2的子串" << endl;

}

else

{

//比较s1和s2中的元素相不相等，相等则判断是不是子串，不相等则s1往后比较

for (int i = 0; i <= b - a; i++)//循环次数

{

int j;

for (j = 0; j < a; j++)//s1

{

if (s1[j] != s2[j + i])

{

break;

}

}

if (j == a)

{

return i;

}

}

return -1;

}

}

int main()

{

char s1[100], s2[100];

cout << "Enter the first string : ";

cin.getline(s1, 100);

cout << "Enter the second string: ";

cin.getline(s2, 100);

if (indexOf(s1, s2) != -1)

{

cout << "indexOf" << "（“";

for (int a = 0; a < strlen(s1); a++)

cout << s1[a];

cout << "”）and（“";

for (int b = 0; b < strlen(s2); b++)

cout << s2[b];

cout << "”）is" << indexOf(s1, s2)+1<<endl;

}

else

{

cout << "indexOf" << "（“";

for (int a = 0; a < strlen(s1); a++)

cout << s1[a];

cout << "”）and（“";

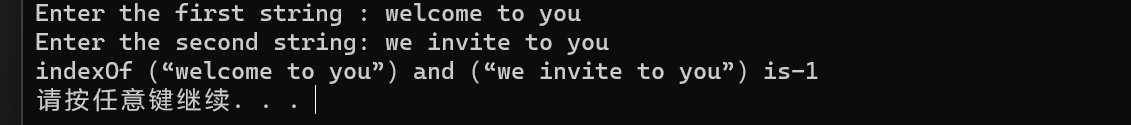
for (int b = 0; b < strlen(s2); b++)

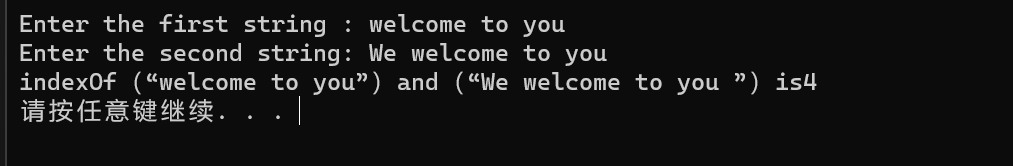
cout << s2[b];

cout << "”）is" << indexOf(s1, s2) << endl;

}

system("pause");

}



题6：

#include"head.h"

void count(const char s[], int counts[])

{

for (int j = 0; j < 26; j++)

{

counts[j] = 0;

}

for (int i = 0; i < strlen(s); i++)

{

char ah;

ah = tolower(s[i]);

if (isalpha(ah))

{

counts[ah - 'a']++;

}

}

for (int j = 0; j < 26; j++)

{

if (counts[j] > 0)

{

char ch = (char)(j + 'a');

cout << ch << " " << counts[j] << " times" << endl;

}

}

}

int main()

{

char s[100];

int counts[26];

cin.get(s, 100);

count(s, counts);

}



指针1：

#include"head.h"

int indexof(const char\* s1, const char\* s2)

{

int a, b;

a = strlen(s1);

b = strlen(s2);

if (a > b)

{

cout << "s1不是s2的子串" << endl;

}

else

{

//比较s1和s2中的元素相不相等，相等则判断是不是子串，不相等则s1往后比较

for (int i = 0; i <= b - a; i++)//循环次数

{

int j;

for (j = 0; j < a; j++)//s1

{

if (s1[j] != s2[j + i])

{

break;

}

}

if (j == a)

{

return i;

}

}

return -1;

}

}

int main()

{

char s1[100];

char s2[100];

cout << "输入第一个字符串：";

cin.getline(s1, 100);

cout << "输入第二个字符串：";

cin.getline(s2, 100);

indexof(s1, s2);

if (indexof(s1, s2) != -1)

{

cout << "indexOf" << "（“";

for (int a = 0; a < strlen(s1); a++)

cout << s1[a];

cout << "”）and（“";

for (int b = 0; b < strlen(s2); b++)

cout << s2[b];

cout << "”）is" << indexof(s1, s2) + 1 << endl;

}

else

{

cout << "indexOf" << "（“";

for (int a = 0; a < strlen(s1); a++)

cout << s1[a];

cout << "”）and（“";

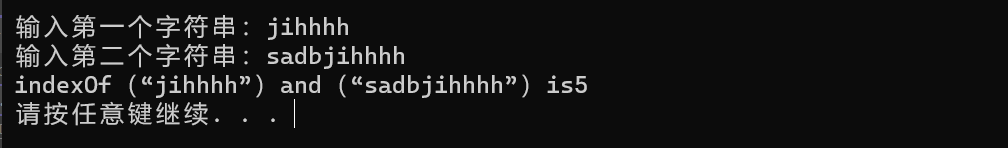
for (int b = 0; b < strlen(s2); b++)

cout << s2[b];

cout << "”）is" << indexof(s1, s2) << endl;

}

system("pause");

}

指针2：

#include"head.h"

#include<cmath>

int parseHex(const char\* const hexString)

{

int aa = strlen(hexString);

int n=0;

for (int i = 0; i < aa; i++)

{

char ch = hexString[i];

if (isdigit(ch))

{

n =n+ (ch-48)\*pow(16, aa - i-1);

}

else if (isalpha(ch))

{

ch = tolower(ch);

switch (ch)

{

case 'a':

n = n + 10\*pow(16, aa - i-1);

break;

case 'b':

n = n + 11\*pow(16, aa - i-1);

break;

case 'c':

n = n + pow(16, aa - i-1);

break;

case 'd':

n = n + pow(16, aa - i-1);

break;

case 'e':

n = n + pow(16, aa - i-1);

break;

case 'f':

n = n + pow(16, aa - i-1);

break;

default:

cout << "输入的字符串不合法" << endl;

}

cout << n << endl;

}

}

cout << n << endl;

return 0;

}

int main()

{

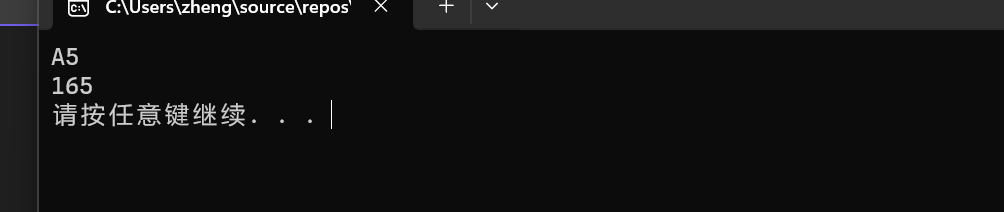
char s[100];

cin.getline(s, 100);

parseHex(s);

system("pause");

}



指针3：

#include"head.h"

void sort(int arr[], int n)

{

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

for (int j = 0; j < n - i - 1; j++)

{

if (arr[j] > arr[j + 1])

{

int tem;

tem = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = tem;

}

}

}

int main()

{

int n;

cout << "输入数组元素的个数：";

cin >> n;

int\* arr = new int[n];

cout << "输入数组元素为：";

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

{

cin >> arr[i];

}

sort(arr,n);

cout << "排列后的数组为：";

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

{

cout << arr[i]<<" ";

}

system("pause");

}

# 遇到的问题与解决办法：

试验总体难度比上次要大很多，所以耗费的时间也比较多。问题主要出在对代码的运用不太熟练，而且题目中的逻辑关系有时会搞混。但最大的问题还是对于bug的查找，这也是实验中最耗费时间的部分。又是因为自己的粗心或者思维上的一时混乱，常常会导致bug的出现。

解决方法主要还是借助互联网查找相关的资料，而对于bug还是得自己耐心查找，如果还有什么问题的话，也会找师兄师姐请教。在与师兄的交流中，也学到了很多。

# 体会：

这次实验收获还是很大的，尤其是在与师兄的交流中学到的关于bug查找的相关技巧，让我受益颇深。还有就是对于代码的熟练的度有了明显的提升。