**华 中 科 技 大 学**

**《计算机基础与程序设计（C++）》**

**实 验 报 告**

**实验名称：第五次上机实验 实验学时：16 姓名：陈淏睿 学号：U201612696 班级：信管（医）1601 实验日期：12.7-12.9 教师批阅签字：**

**一、实验目的**

* 掌握字符型数组、字符型指针和数值型数组的异同 。
* 掌握自定义函数的定义，声明和调用。
* 掌握函数参数的值传递和地址传递的区别。
* 掌握递归算法的意义。
* 明确变量的作用域和生存期的意义。

**二、实验内容**

**1．阅读程序**

1）#include <iostream>

#include<cmath>

using namespace std;

int main()

{ int a=10, \*p=&a;

char ch=’a’, \*q=&ch;

char str[ ]=”abcd”,\*qq=str;

cout<<\*p<<endl;

cout<<\*q<<endl;

cout<<\*qq<<endl;

cout<<p<<endl;

cout<<q<<endl;

cout<<qq<<endl;

return 0;

}

**运行上述的程序，并解释每个输出结果。**

1.通过指针\*p输出其指向的数10；2.通过指针\*q输出其指向的字符a；3.qq内存储字符串abcd的首地址，通过指针\*qq输出字符串的首字符a；4.通过p输出为变量a分配的地址；5.通过q对从首地址内字符开始的字符串整体访问，但其为单个字符，因此在访问之后的地址时输出乱码；6.通过qq对从首地址内字符开始的字符串整体访问，碰到’\0’结束。

2）#include <iostream>

#include<cmath>

using namespace std;

int main()

{ int age ;

char name[11];

cin>>age；

cin.get () ;

cin.getline(name,11);

cout<<age<<endl;

cout<<name<<endl;

return 0; }

**运行上述的程序，其结果是否正确，如不确请修改正确，并解释原因。**

不正确，在age后再输入数字不能正确区分其属于age的一部分还是属于字符串。修改措施：在两次输入中间插入cin.get() 人工隔离两次输入的内容。

3） #include <iostream>

#include <cstring>

using namespace std;

int main( )

{

char rstr[10 ]="abcdefgh", dstr[10];

char \* p=rstr, \*q=dstr;

strcpy(dstr,rstr);

cout<<"输出字符串rstr："<<rstr<<endl;

cout<<"输出字符串dstr："<<dstr<<endl;

cout<<"输出字符串p："<<p<<endl;

cout<<"输出字符串q："<<q<<endl;

return 0;

}

**阅读上述程序并运行，分析其结果。若将上述程序修改如下，重新运行，结果如何？请解释原因。**

int main( )

{

char \*rstr="abcdefgh", ~~\*dstr~~ dstr[9];

strcpy(dstr,rstr);

cout<<"输出字符串rstr："<<rstr<<endl;

cout<<"输出字符串dstr："<<dstr<<endl;

return 0;

}

**上述程序的功能是将字符串rstr复制到目标串dstr中，运行该程序结果是否正确，如不正确请修改正确，并解释原因。**

\*rstr和\*dstr均为字符指针，只能进行输出操作，不可对其进行再赋值，因此程序报错。将被赋值数组改为字符数组的形式，即可对其进行strcpy复制操作。

1. **编程题**

1）编写两个函数，分别求两个整数的最大公约数和最小公倍数，在主函数中调用这两个函数，计算两个数的最大公约数和最小公倍数并输出结果。

**要求：**

1. 程序原代码。**（直接粘贴在此）**

#include < iostream >

using namespace std ;

//1.辗转相除

int maxdiv ( int a , int b )

{

int c , d , temp ;

c = ( a >= b ) ? a : b ;

d = ( a < b ) ? a : b ;

while ( c % d )

{

c %= d ;

temp = c ;

c = d ;

d = temp ;

}

return d ;

}

//2.辗转相减

int maxdiv ( int a , int b )

{

int c , d , temp ;

c = ( a >= b ) ? a : b ;

d = ( a < b ) ? a : b ;

while ( c != d )

{

c -= d ;

if ( c < d )

{

temp = c ;

c = d ;

d = temp ;

}

}

return d ;

}

int minmul ( int a , int b )

{

int i , c = ( a >= b ) ? a : b ;

for ( i = c ; ; i ++ )

{

if ( i % a == 0 && i % b == 0 )

break ;

}

return i ;

}

int main()

{

int a , b ;

cout << "请输入a , b的值：" ;

cin >> a >> b ;

cout << "最小公倍数为：" << minmul ( a , b ) << endl ;

cout << "最大公约数为：" << maxdiv ( a , b ) << endl ;

return 0;

}

1. 解决本题的算法思路描述

1.单独调出两个函数操作最大值与最小值；2.最大公约数用辗转相除，最小公倍数用循环求出，当其对两个数取余均为零时满足条件，返回值；

1. 调试过程的记载（包括出现的错误，以及修改）

1在做第一次取余后错误地将c , d循环赋值给a和b；2.最大公约数中对c和d先后用( c >= d ) ? c : d; ( c < d ) ? c : d时忘了c在进行运算后的值已改变，在d中的值为新c值

1. 归纳本题应用的知识点或算法

1.函数嵌套；2.辗转相除；3.循环

**2）**编写程序随机生成50个100-200之间的随机整数，找出其中的素数。**要求判断素数的功能利用函数实现**，在main函数中调用该函数输出所有的素数。

**要求：**

1. 程序原代码。**（直接粘贴在此）**

# include < iostream >

# include < ctime >

# include < cmath >

# include < cstdlib >

using namespace std ;

int count = 0 ;

void judge ( int n )

{

int i , flag = 0 ;

for ( i = 2 ; i <= sqrt ( n ) ; i ++ )

if ( n % i == 0 )

flag = 1 ;

if ( flag == 0 )

{

cout << n << ' ' ;

count ++ ;

if ( ! ( count % 5 ) )

{

cout << endl ;

count = 0 ;

}

}

}

int main()

{

int a [ 50 ] , i , count = 0 ;

cout << "输出50个随机数：" << endl ;

for ( i = 0 ; i < 50 ; i ++ )

{

srand ( ( unsigned ) time ( 0 ) + i ) ;

a [ i ] = 100 + ( int ) ( rand() % 100 ) ;

cout << a [ i ] << ' ' ;

count ++ ;

if ( ! ( count % 10 ) )

{

cout << endl ;

count = 0 ;

}

}

cout << "输出其中的素数：" << endl ;

for ( i = 0 ; i < 50 ; i ++ )

judge ( a [ i ] ) ;

cout << endl ;

return 0;

}

1. 解决本题的算法思路描述

1.判断素数程序模板；2.计数器控制换行；3.若局部变量与全局变量重名则全局变量在函数内被屏蔽；4.随机数种子生成随机变量

1. 调试过程的记载（包括出现的错误，以及修改）

1.被调函数内的计数器位置错误导致换行失效；2.将输出素数的条件与标志变量的关系搞反；

1. 归纳本题应用的知识点或算法

1.调用函数；2.判断素数模板；3.计数器

**3）**编写一个函数实现将一个十六进制整数转换为一个十进制整数。例如，输入”A2”转换为162。

提示：循环对读入的每个字符转换成对应的十进制数字，比如’1’的值为1，’A‘为10，’F’为15，将转换后得到的数字进行计算处理。

**函数的原型可以声明为：**

int funChange(char s[]); //函数的功能是将s数组存放的字符串转换为十进制数值返回，参数应采用数组名传递方式，即形参数组与实参数组共用同一地址空间。

**要求：**

1. 程序原代码。**（直接粘贴在此）**

# include < iostream >

using namespace std ;

int funChange ( char s [ ] , int n )

{

int dec , sum = 0 , i ;

i = 0 ;

while ( s [i] != '\0' )

{

if ( s [i] >= '0' && s [i] <= '9' )

{

dec = s [i] - '0' ;

}

else

if ( s [i] >= 'A' && s [i] <= 'F' )

dec = s [i] - 'A' + 10 ;

if ( s [i] >= 'a' && s [i] <= 'f' )

dec = s [i] - 'a' + 10 ;

sum = sum \* 16 + dec ;

i ++ ;

}

cout << sum ;

return sum ;

}

int main()

{

int dec ;

char model [20] ;

cout << "请输入字符串，注意其中只能含有数字及a-f内的字母（不区分大小写）,不可含有空格\n" ;

cin.getline ( model , 20 ) ;

dec = funChange ( model , 20 ) ;

cout << "相应的十进制数为" << dec << endl ;

return 0;

}

1. }解决本题的算法思路描述

1.独立函数完成从字符数组向数字数组的转换；2.while循环判断字符串是否终止；3.十六进制数转十进制数的数学操作

1. 调试过程的记载（包括出现的错误，以及修改？）

1.转换程序用for语句循环时字符串到头后依旧进行循环导致输出数据乱码

1. 归纳本题应用的知识点或算法

1.字符转数字；

**4 ）**编写函数验证哥德巴赫猜想，任意一个充分大的偶数均可表示成两个素数之和。要求定义两个函数，一个函数判断一个整数是否是素数，另一个函数验证哥德巴赫猜想，既将一个偶数分解为两个素数，并返回这两个素数。利用函数的嵌套调用完成（即在验证哥德巴赫猜想的函数中调用判断素数函数）。例如 8=3+5这样的显示信息在主函数中完成。

**要求：**

1. 程序原代码。**（直接粘贴在此）**

# include < iostream >

# include < math.h >

using namespace std ;

int judge ( int n )

{

int i , flag = 1 ;

for ( i = 2 ; i <= sqrt ( n ) ; i ++ )

if ( ! ( n % i ) )

{

flag = 0 ;

break ;

}

return flag ;

}

int gede ( int aim , int &i )

{

for ( i = 2 ; i <= aim / 2 ; i ++ )

if ( judge ( i ) && judge ( aim - i ) )

{

return 1 ;

break ;

}

return 0 ;

}

int main()

{

int aim , i = 0 ;

cout << "请输入待判断偶数\n" ;

cin >> aim ;

if ( gede ( aim , i ) )

cout << aim << '=' << i << '+' << aim - i << endl ;

return 0;

}

1. 解决本题的算法思路描述

1.循环结构判断素数；2.两个素数是验证哥德巴赫猜想的充要条件，故if内条件应为对两个数的函数处理值均为真；3.哥德巴赫猜想成立后应至少返回一个素数，此时运用引用传递。

1. 调试过程的记载（包括出现的错误，以及修改？）

未出现错误。

1. 归纳本题应用的知识点或算法

1.函数嵌套；2.引用传递；3.求素数模板；4.状态变量

**5）** 编写函数，求出任意一个一维数组元素中的最大值和最小值的下标。要求在主函数中输入数组元素的值，输出最大值和最小值。(最大最小值只需考虑一个即可)

**提示：**本题要求将一维数组的元素传入到函数进行处理，对于大量的数据的传递，最好的方式是使用数组名作为实际参数传递，在这种情况下，形参可以是指针也可以是数组，通过形参能直接对实参数组的数据进行处理。本题的问题是获取数组元素的最大值和最小值，在函数中有两个值需要返回到主函数，因此不能用return语句，需使用指针或引用参数进行回传。

**函数的原型可以声明为：**

void funMaxMin(int a[], int n, int \*max, int \*min) //函数的功能获取数组元素的最大值和最小值下标。

**要求：**

1. 程序原代码。**（直接粘贴在此）**

# include < iostream >

using namespace std ;

void findmax ( double judge [] ,int n , double &max , int &submax )

{

int i ;

max = judge [0] ;

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

if ( judge [i] > max )

{

max = judge [i] ;

submax = i ;

}

}

void findmin ( double judge [] ,int n , double &min , int &submin )

{

int i ;

min = judge [0] ;

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

if ( judge [i] < min )

{

min = judge [i] ;

submin = i ;

}

}

int main()

{

int i , submax = 0 , submin = 0 ;

double origin [20] , max , min ;

cout << "输入原数组：" ;

for ( i = 0 ; i < 20 ; i ++ )

{

cin >> origin [i] ;

}

findmax ( origin , 20 , max , submax ) ;

findmin ( origin , 20 , min , submin ) ;

cout << "最大值为：" << max << '\t' << "其下标为：" << submax << endl ;

cout << "最小值为：" << min << '\t' << "其下标为：" << submin << endl ;

return 0;

}

1. 解决本题的算法思路描述

1.对数组整体进行取最大值及下标运算；

1. 调试过程的记载（包括出现的错误，以及修改？）

1.不会写数组传递的格式

1. 归纳本题应用的知识点或算法

1.数组传递；2.引用传递；3.求最值模板

**6）**编写一个递归函数，统计任意位正整数的位数，并在主函数中输入这个整数和输出统计的结果。

**要求：**

1. 程序原代码。**（直接粘贴在此）**

# include < iostream >

using namespace std ;

int stat ( int num , int count )

{

if ( num == 0 )

return count ;

count ++ ;

stat ( ( int ) num / 10 , count ) ;

}

int main()

{

char j ;

int num , count ;

do

{

count = 0 ;

cin >> num ;

cout << stat ( num , count ) << endl ;

cout << "是否继续输入？是-输入Y；否-输入其他任意键\n" ;

cin >> j ;

}

while ( j == 'y' || j == 'Y' ) ;

return 0 ;

}

1. 解决本题的算法思路描述

1.递归调用直到目标数被修改为0；2.计数器统计刨去末位数字的次数

1. 调试过程的记载（包括出现的错误，以及修改？）

1.递归函数里递归终止条件不清楚、不会写；2.计数器误放在递归语句之后导致失灵；3.主函数循环之后不会把计数器归零

1. 归纳本题应用的知识点或算法

1.递归调用；2.计数器

7）编写程序要求能对任意一数值型数组排序，然后用折半查找的方法查询需查找的数。要求排序和查找功能分别用自定义函数实现。**要求：是否找到的信息只能在主函数中输出显示。**

**要求：**

1. 程序原代码。**（直接粘贴在此）**

# include < iostream >

using namespace std ;

void senum ( double a [] , int n , double target , int &flag , int &mid )

{

int high = 4 , low = 0 ;

mid = ( high + low ) / 2 ;

while ( a [mid] != target && low <= high )

{

if ( a [mid] < target )

low = mid + 1 ;

else

high = mid - 1 ;

mid = ( high + low ) / 2 ;

}

if ( a [mid] == target )

flag = 1 ;

}

void size ( double a [] , int n )

{

double temp ;

int noswap , j , i ;

for ( j = 0 ; j < n ; j ++ )

{

noswap = 0 ;

for ( i = 4 ; i > j ; i -- )

if ( a [i] < a [i-1] )

{

temp = a [i] ;

a [i] = a [i-1] ;

a [i-1] = temp ;

noswap = 1 ;

}

if ( noswap == 0 )

break ;

}

}

int main()

{

int i , flag = 0 , mid ;

double a [5] , target ;

cout << "输入原数组:\n" ;

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

cin >> a [i] ;

size ( a , 5 ) ;

cout << "输出排序后数组：\n" ;

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

cout << a [i] << ' ' ;

cout << endl << "输入待查找数:\n" ;

cin >> target ;

senum ( a , 5 , target , flag , mid ) ;

if ( flag == 1 )

cout << "找到！目标数下标为" << mid << endl ;

else

cout << "未找到目标数！\n" ;

return 0 ;

}

1. 解决本题的算法思路描述

1.状态变量决定输出内容；2.检查数组是否成功排序

1. 调试过程的记载（包括出现的错误，以及修改？）

1.没有想清楚折半查找程序里要处理哪些变量；2.数组传递格式不清楚

1. 归纳本题应用的知识点或算法

1.冒泡排序模板；2.折半查找模板；3.引用传递；

**三、对本次实验内容及方法、手段的改进建议，以及实验心得**

**实验心得包括**：

1）哪些知识点已掌握 除了下面那些都掌握了

2）哪些知识点有困难 数组传递，多参数混合引用，递归调用，字符指针的操作范围

3）对讲课的建议 想要加几节课复习重点或者讲例题

4）对没有掌握知识的补救建议 多练