# 实验4 函数

## 一. 实验目的

(1) 掌握函数的定义及其调用，变量存储特性，作用域。

(2) 掌握函数的定义及其调用。

(3) 掌握C++函数参数的传递的定义和应用。

(4) 掌握自动变量和静态变量的概念。

(5) 掌握函数原型、块、函数和文件作用域的概念。

(6) 掌握全局变量与局部变量的概念。

(7) 了解、掌握多文件程序上机调试的过程。

## 二. 实验内容及步骤

2. 上机调试下面程序，改正其不合理之处。

#include <iostream>

using namespace std;

main()

{

int x,n,s;

s=power(x,n);

}

power(y)

{

int i,p=1;

for(i=1;i<=n;i++)

p=p\*y;

}

3. 按照要求完成以下程序

1) 计算并输出 500 以内的所有“亲密数”对，并输出所有“完数”之和。

提示：6的因子为2和3。如果自然数M的所有因子（包括1但不包括自身，下同）之和为N，而N的所有因子之和为 M，则称M与N为一对“亲密数”。例如，220的所有因子之和为：1+2+4+5+10+11+20+22+44+55+110=284，而284的所有因子之和为：1+2+4+71+142=220，因此220与 284 为一对“亲密数”。

如果一个自然数的所有因子之和恰好等于它自身，则称该自然数为“完数”。例如，6不仅与它自身构成一对“亲密数”，且6 是一个“完数”，即6的所有因子之和为：1+2+3=6。

具体要求：

① 编写一个函数 fsum（n），返回给定正整数 n 的所有因子之和。

② 编写一个主函数，调用函数fsum（n），寻找并输出500以内的所有“亲密数”对以及计算所有“完数”之和。

③ 在输出每队“亲密数”时，要求小数在前、大数在后，并去掉重复的数对。例如，220和284是一对“亲密数”，而284和220也是一对“亲密数”，此时只需输出220与284这对“亲密数”。

④ 输出要有文字说明（英文或汉语拼音）。输出时每对“亲密数”用一对方括号括起来，两数之间用逗号分隔，且所有的“亲密数”对占一行。输出形式为：

各对“亲密数”

“完数”之和

⑤ 在程序内部添加必要的注释（至少有 3 处）。

⑥ 将两个函数放在一个文件中进行编译、连接并运行。

⑦ 将两个函数分别在三个文件中进行编译、连接并运行。

实验步骤：

① 建立新的项目文件。

② 在项目中建立新的C源程序文件。

③ 按要求⑥完成编译和连接。在空白的源程序文件中输入主函数程序代码。程序代码如下所示：

#include <iostream>

using namespace std;

void main()

{

int facsum(int); /\*函数原型\*/

int m,n,s=0;

cout<<"qinmishudui:\n";

for (m=2;m<=500;m++)

{

n=facsum(m); /\*函数调用\*/

if((m==facsum(n))&&(m<=n)) cout<<"["<<m<<","<<n<<"]"<<endl; /\*按要求输出亲密数对\*/

if(n==m) s=s+m;

}

cout<<"wanshuzihe:"<<s<<"\n";

}

int facsum(int m1) /\*函数定义facsum，求m的所有因子之和\*/

{

int f=1,k;

for(k=2;k<=m1/2;k++)

if(m1%k==0)

f=f+k;

return f;

}

④按要求⑦完成编译和连接。程序代码如下所示：

//mymain.cpp

#include <iostream>

#include " myheader.h"

using namespace std;

void main()

{

int m,n,s=0;

cout<<"qinmishudui:\n";

for (n=2;n<=500;n++)

{

m=facsum(n); /\*函数调用\*/

if((n==facsum(m))&&(n<=m)) cout<<"["<<n<<","<<m<<"]\n"; //输出亲密数对

if(m==n) s=s+n;

}

cout<<"wanshuzihe:"<<s<<endl; /\*按要求输出完数之和\*/

}

//myheader.h

int facsum(int n); /\*函数原型\*/

//myfac.cpp

int facsum(int n) /\*定义函数 facsum，求 n 的所有因子之和\*/

{

int f=1,k;

for(k=2;k<=n/2;k++)

if(n%k==0) f=f+k;

return f;

}

⑤ 源程序编译连接，若有语法错误，则根据提示对程序相应代码进行修改，然后再一次编译连接，重复此过程直到程序无语法错误。

⑥ 运行程序，如发现错误进行修改，直到运行出正确结果。

2) 求两个整数的最大公约数和最小公倍数，用一个函数求最大公约数。用另一函数根据求出的最大公约数求最小公倍数。

具体要求：

① 不用全局变量，分别用两个函数求最大公约数和最小公倍数。两个整数在主函数中输入，并传给求最大公约数函数，求出的最大公约数返回主函数，然后再与两个整数一起作为实参传递给求最小公倍数函数，以求出最小公倍数，返回到主函数输出最大公约数和最小公倍数。

② 用全局变量的方法，分别用两个函数求最大公约数和最小公倍数，但其值不由函数带回。将最大公约数和最小公倍数都设为全局变量，在主函数中输出它们的值。需要注意的是，不提倡在程序中过多使用全局变量。

实验步骤：

① 建立新的项目文件。

② 在项目中建立新的 C源程序文件。

③ 在空白的源程序文件中输入程序代码。程序代码如下所示：

a. /\*求最大公约数和最小公倍数\*/

#include <iostream>

using namespace std;

void main()

{

int hcf(int,int); /\*说明最大公约数 hcf函数\*/

int lcd(int,int,int); /\*说明lcd 函数\*/

int m,n,h,l;

cout<<"please input m,n:";

cin>>m>>n;

h=hcf(m,n);

cout<<"H.C.F="<<h<<endl;

l=lcd(m,n,h);

cout<<"L.C.D="<<l<<endl;

}

int hcf(int m,int n) /\*定义求最大公约数的函数\*/

{

int t,r;

if(n>m)

{

t=n;n=m;m=t;

}

while ((r=m%n)!=0)

{

m=n;

n=r;

}

return (n);

}

int lcd(int m,int n,int h) /\*定义求最小公倍数函数\*/

{

return (m\*n/h);

}

b. /\*用全局变量求最大公约数和最小公倍数\*/

#include <iostream>

using namespace std;

int h,l; /\*定义全局变量\*/

void main()

{

void hcf(int,int); /\*说明 hcf 函数\*/

void lcd(int,int); /\*说明 lcd函数\*/

int m,n;

cout<<"please input m,n:";

cin>>m>>n;

hcf(m,n);

lcd(m,n);

cout<<"H.C.F="<<h<<endl;

cout<<"L.C.D="<<l<<endl;

}

void hcf(int m,int n) /\*定义求最大公约数的函数\*/

{ int t,r;

if(n>m)

{

t=n;n=m;m=t;

}

while ((r=m%n)!=0)

{

m=n;

n=r;

}

h=n;

}

void lcd(int m,int n) /\*定义求最小公倍数函数\*/

{

l=m\*n/h;

}

④ 源程序编译连接，若有语法错误，则根据提示对程序相应代码进行修改，然后再一次编译连接，重复此过程直到程序无语法错误。

⑥ 运行程序，如发现错误进行修改，直到运行出正确结果。

4. 完成以下程序，并调试运行

1) 试分析以下函数的功能。

int cmp(int x, int y)

{

int a;

if (x>y) a=1;

else

if (x==y) a=0;

else a=-1;

return (a);

}

实验指导：

首先分析函数的功能，然后用程序运行的方法来验证分析的结果。用程序运行方法验证必须编写一个正确的源程序，请同学自己加上主函数，有主函数调用此函数来验证函数的功能。

2) 阅读下列程序，写出执行结果，然后上机调试程序，比较结果的正确性，从而理解块作用域的概念。

#include <iostream>

using namespace std;

void main()

{

int a=1,b=2,c=3;

cout<<"a="<<a<<"\t"<<"b="<<b<<"\t"<<"c="<<c<<endl;

{

int b=4,c=5;

cout<<"a="<<a<<"\t"<<"b="<<b<<"\t"<<"c="<<c<<endl;

a=b;

{

int c=b;

cout<<"a="<<a<<"\t"<<"b="<<b<<"\t"<<"c="<<c<<endl;

}

cout<<"a="<<a<<"\t"<<"b="<<b<<"\t"<<"c="<<c<<endl;

}

cout<<"a="<<a<<"\t"<<"b="<<b<<"\t"<<"c="<<c<<endl;

}

3) 分析下列程序，写出执行结果，然后上机调试程序，比较结果的正确性，从而理解全局变量和局部变量的概念。

#include <iostream>

using namespace std;

int a=1,b=2;

int f1(int x)

{

int i,b=3;

i=(a\*b)/x;

cout<<"i="<<i<<endl;

return i;

}

int c=3;

int f2(int x,int y)

{

int u,v,a=5;

u=x+y;

v=(b-a)/c;

cout<<"u="<<u<<endl;

cout<<"v="<<v<<endl;

return u-v;

}

void main()

{

cout<<f1(1)+a-b<<endl;

cout<<f2(2,21)+c<<endl;

}

4) 以下程序的运行结果是\_\_\_\_\_\_\_\_，进行上机验证。

#include <iostream>

using namespace std;

void main()

{

increment();

increment();

increment();

}

void increment()

{

int x=0;

x+=1;

cout<<x;

}

5) 以下程序的运行结果是\_\_\_\_\_\_\_\_，进行上机调试验证。

#include <iostream>

using namespace std;

void main()

{

int a=1,b=2,c;

c=max(a,b);

cout<<"max is "<<c<<endl;

}

int max(int x, int y)

{

int z;

z=(x>y)?x:y;

return(z);

}

6) 在下面程序中的空格中填入适当内容，使得运行结果是A+B=9 。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int a=5; int b=7;

void main()

{

int \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

c=plus(a,b);

cout<<"A+B="<<c<<endl;

}

int plus(int x,int y)

{

int z;

z=x+y;

return \_\_\_\_\_\_;

}

7) 以下程序的功能是根据输入的"y"("Y")与"n"("N")，在屏幕上分别显示出"This is YES."与"This is NO."。请填空，进行上机验证。

#include<iostream>

using namespace std;

void YesNo(char ch)

{

switch(ch)

{

case 'y':

case 'Y': cout<<"\nThis is YES.\n"; \_\_\_\_\_\_\_\_;

case 'n':

case 'N': cout<<"\nThis is NO.\n";

}

}

void main()

{

char ch;

cout<<”Enter a char 'y','Y' or 'n','N': ";

\_\_\_\_\_>>ch;

YesNo(ch);

}

8) 使用递归函数，把任意一个正整数颠倒过来输出。例如，输入正整数678，输出应为876。

9) 找出所有各位数字之和为15的三位数, 并统计这样的三位数的个数

提示：设计一函数，判断一个三位数的各位数字之和是否为15，如果是，则返回1，否则返回0。

10) 从键盘上输入五个整数分别存放到变量a，b，c，d，e中，将这五个数按升序排列并且输出。

实验指导：

① 将a，b，c，d，e进行升序排列的算法可以用：

从a，b，c，d，e中找出最小值，用对调法使其置于变量a；

从b，c，d，e中找出最小值，用对调法使其置于变量b；

从c，d，e中找出最小值，用对调法使其置于变量c；

从d，e中找出最小值，用对调法使其置于变量d；

② 输入输出时均需加上提示信息，才能在运行程序时有清晰的说明。

11) 编写程序，给学生出一道加法运算题，然后判断学生输入的答案对错与否。按下列要求以循序渐进方式编程。

程序一：让学生自己输入两个加数，通过这两个加数给学生出一道加法运算题，如果学生输入答案正确，则显示“Right！”否则显示“Not correct! Try again!”，程序结束。

程序二：让学生自己输入两个加数，通过这两个加数给学生出一道加法运算题，如果学生输入答案正确，则显示“Right！”否则显示“Not correct! Try again!”，直到做对为止。

程序三：让学生自己输入两个加数，通过这两个加数给学生出一道加法运算题，如果学生输入答案正确，则显 示“Right！”否则提示重做，显示“Not correct! Try again!”，重做机会最多给三次，如果三次仍未做对，则显示“Not correctm,You have tried three times! Test over!”，程序结束。

程序四：将程序设计成连续做十道题，让学生自己输入两个加数，通过这两个加数给学生出一道加法运算题，如果学生输入答案正确，则显示“Right！”否则显示“Not correct！”，不给机会重做，十道题做完后，按每题10分统计总得分，然后打印出总分和做错的题数。

将上述渐进方式的编程改为完成减法、乘法、除法等功能，程序应该怎样修改？试一试。

5!+7!

8!

y=

12) 使用多文件结构编写程序计算 和 的值，提示：n!=1\*2\*3\*……\*n。要求：

s=1!+2!+……+10!

① 把求阶乘的函数存放在文件file1.cpp中；

② 把求 的函数存放在文件file2.cpp中；

5!+7!

8!

y=

③ 求 的函数存放在文件file3.cpp中；

s=1!+2!+……+10!

④ 函数原型在头文件headf.h中声明；

⑤ 主函数存放在文件mainf.cpp中。

上机调试程序，从而掌握VC6多文件程序上机调试的操作过程。

13) 编写程序验证哥德巴赫猜想：一个不小于6的偶数可以表示为两个素数之和，如6=2+3，8=3+5，10=3+7，…。

实验指导：

假设输入不少于6的偶数为n，n=a+b，分别判断a和b是否为素数，如果两者均为素数，则满足要求，输出结果。如果两者并非同为素数，则不满足要求，此时改变a和b的值，重新进行测试。先设a的值为3(3是除了2以外最小的素数。a的值不应设为2，因为b是偶数，若a是偶数，则b必为偶数而不是素数)，则b=n-a，先判别a是否素数，再判别b是否也是素数，如果是则输出；然后改变a、b的值，使a加2，b=n-a，重新进行测试。如此一组一组地测试，直至a的值从3到a≤n/2为止，不用3到到n。

14) 鸡兔同笼, 鸡和兔的头共有h个，腿共有g条。输入h和g，用循环结构求出鸡兔各多少只？

实验指导：

一只鸡有2条腿，而一只兔子有4条腿。此题会有无解的时候。

使用循环结构编程时，按照“一只鸡有2条腿，而一只兔子有4条腿”来设置循环条件。

16) 编写程序：判别一个输入的整数数组中各元素的值，若大于0 则输出该值，若小于等于0则输出0。

实验指导：本程序中可先定义一个无返回值函数用以判断各元素的值，并说明其形参为整型变量，在函数体中根据形参的值输出相应的结果。在main函数中定义一个有若干元素的数组，用循环控制语句输入数组各元素，每输入一个就以该元素作实参调用一次函数，即把元素的值传送给形参，供函数使用。

17) 编写一个函数,在一个数组中查找出其值等于给定值x的第一个元素,如果查找成功, 返回这个元素的下标, 否则返回-1。

18) 编写一个函数, 在一个数组中查找出其值等于给定值x的所有元素,函数的返回值是数组中其值等于给定值的元素的个数。

19) 定义函数Max\_Array 求一维数组中的最大元素值。尽可能设计不同形式参数的函数原型，并定义它们的实现。用main函数进行测试，比较不同参数的定义和使用形式。

20) 排序是为了快速查找。定义一个函数，在有序数组中查找指定值，想一想怎样才能找的更快。