

C++库函数

- C++的系统库中提供了几百个函数可供程序员使用。

例如：求平方根函数（sqrt）、求绝对值函数（abs）等。

- 使用库函数时要包含相应的头文件，**cmath.h**

```
#include <cmath>
using namespace std;
```

函数	语法	功能	返回值类型	运用举例
幂	pow(x, y)	x的y次幂	double	pow(2,5)=32
平方根	sqrt(x)	x的平方根	double	sqrt(16)=4
绝对值	abs(x)	整数的绝对值	int	abs(4.6) = 4 abs(0) = 0 abs(-5) = 5
	fabs(x)	实数的绝对值	double	fabs(-9.6) = 9.6 fabs(0.34) = 0.34
指数函数	exp(x)	e^x	double	exp(1) = 2.71828 exp(2) = 7.38906
自然对数	log(x)	x的自然对数 (底为e)	double	log(2.718282) = 1.0 log(7.389056) = 2.0
对数	log10(x)	x的对数 (底为10)	double	log(10.0) = 1.0 log(100.0) = 2.0
正弦	sin(x)	x的正弦值	double	sin(60*3.141593/180)=0.866025
余弦	cos(x)	x的余弦值	double	cos(60*3.141593/180)=0.5
正切	tan(x)	x的正切值	double	tan(60*3.141593/180)=1.73205

- rand

函数原型：**int rand();**
所需头文件：**<cstdlib>**

随机数产生器
(rand()函数)

功能和返回值：求出并返回一个伪随机数。在使用rand函数时，如果没有初始化随机数产生器，则每次程序执行将初始化同样的一系列数字（伪随机数）。要产生真正的随机数必须在使用随机数产生器的程序中初始化该随机数产生器。

- srand

函数原型：**void srand(unsigned int seed);**
参数：**seed**产生随机数的种子。
所需头文件：**<cstdlib>**

功能：为使rand()产生一序列随机整数而设置起始点。一般使用系统内置函数time()作为种子值，该函数包含在头文件<ctime>中。

srand(time(NULL)); //利用系统时间初始化随机数产生器

- 产生特殊范围内的随机数字，可使用如下公式

下限+rand()%(上限-下限+1)

下限是该范围的最小值，上限是该范围的最大值。

实验7-1

1、实现功能：根据键盘输入的数 x ，计算分段函数 $f(x)$ 的值。

$$y = f(x) = \begin{cases} x \text{的平方根} & (x > 0) \\ 0 & (x = 0) \\ x \text{的绝对值} & (x < 0) \end{cases}$$

(1) 主函数功能：从键盘输入数 x ，通过调用子函数得到计算结果并输出。

(2) 子函数功能：定义有返回值函数，实现计算该分段函数值的功能。

要求：求平方根和绝对值使用数学函数 $\text{sqrt}(x)$ 、 $\text{fabs}(x)$ 。

```
#include <iostream>
#include <cmath>
using namespace std;
void main( )
{
```

$$y = f(x) = \begin{cases} x \text{ 的平方根} & x > 0 \\ 0 & x = 0 \\ x \text{ 的绝对值} & x < 0 \end{cases}$$

```
double fun_F(double x); //子函数声明
```

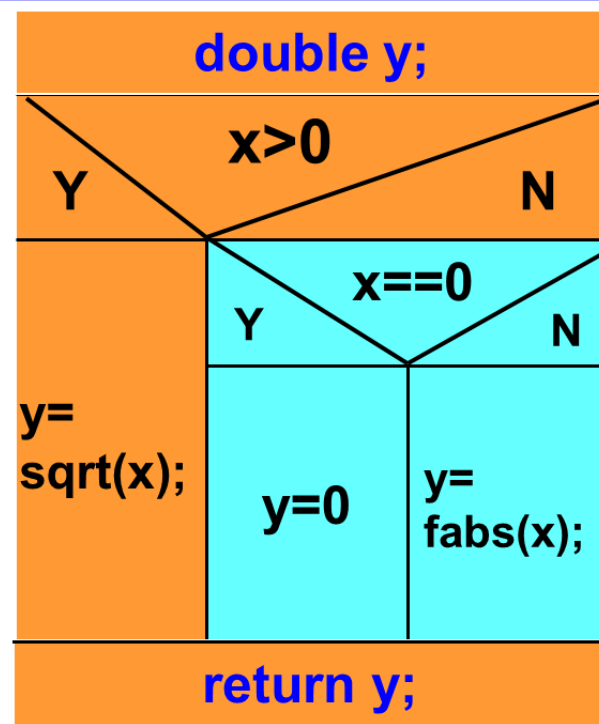
```
double x,y;
cout<<"请输入x的值: ";
cin>>x;
    y=fun_F(x);           //函数调用
cout<<"y的值为: "<<y<<endl;
}
```

子函数定义:

```
double fun_F(double x)
{
    ..... //函数体
}
```

子函数声明:

```
double fun_F(double x);
```



7-2 实现功能：1~10加减乘运算训练系统。

(1) **主函数功能**：随机产生一个1~10加减乘算术式（数字在1~10之间随机产生，运算符在加减乘中随机产生，如2+3=、8-4=、6*3=），要求用户输入计算结果，通过调用子函数得到正确答案，判断用户输入的计算结果是否正确并输出结论。

(2) **子函数功能**：定义有返回值函数，实现加减乘算术式的计算功能。

下限+rand()%(上限-下限+1)

int a,b,m,s,s1; char op;

产生两个随机数a和b，和运算符op，并输出运算式

调用子函数得到运算结果s

由用户计算得到运算式的结果s1

T s1==s F

输出“计算正确”

输出“计算错误”

```
int a,b,m,s; char op;
srand(time(NULL));
a=1+rand( )%(10-1+1);
b=1+rand( )%(10-1+1);
m=1+rand( )%(4-1+1);
switch(m)
{
case 1: op='+'; break;
case 2: op='-'; break;
case 3: op='*'; break;
case 4: op='/'; break;
}
s=calculation(a,b,op); //函数调用
```

```
int a,b,m,s,s1; char op;
```

产生两个随机数a和b, 和运算符op, 并输出运算式

调用子函数得到运算结果s

由用户计算得到运算式的结果s1

T	s1==s	F
输出“计算正确”		输出“计算错误”

子函数定义:

```
int calculation(int a,int b,char op)
{
    int result;
    switch(op)
    {
        case '+': cout<<a<<'+<<b<<='=';
                    result=a+b;break;
        case '-': cout<<a<<'-<<b<<='=';
                    result=a-b; break;
        case '*': cout<<a<<'*<<b<<='=';
                    result=a*b; break;
        case '/': cout<<a<<'/<<b<<='=';
                    result=a/b; break;
    }
    return result;
}
```

```
int a,b,m,s,s1; char op;
```

```
srand(time(NULL));
```

```
a=1+rand( )%(10-1+1);
```

```
b=1+rand( )%(10-1+1);
```

```
m=1+rand( )%(4-1+1);
```

```
switch(m)
```

```
{
```

```
case 1: op='+';break;
```

```
case 2: op='-'; break;
```

```
case 3: op='*'; break;
```

```
case 4: op='/'; break;
```

```
}
```

```
s=calculation(a,b,op); //函数调用
```

```
cout<<“请输入 计算结果”<<endl;
```

```
cin>>s1;
```

```
if(s==s2)
```

```
    cout<<“计算正确! ”<<endl;
```

```
else
```

```
    cout<<“计算错误! ”<<endl;
```

子函数声明:

```
int calculation(int a,int b,char op);
```

7-3 实现功能：计算a的b次方与b的a次方的和。

(1) **主函数功能**：从键盘输入两个整数a、b ($a>0$, $b>0$)，通过调用子函数计算a的b次方与b的a次方的和并输出结果。

(2) **子函数功能**：定义有返回值函数，实现计算a的b次方的功能。

如何定义函数？

要处理的数据是x，因此必须有一个**参数x**，类型为**double**。另外，需求x的n次幂，故还应有一个**参数n**，类型为**int**。**返回值为double类型**。即

输入a,b

调用子函数
求a的b次幂
和b的a次幂

输出结果

```
double power(double x, int n)
{
    .....
}
```


power函数

```
double power(double x, int n);
```

$\text{val} = x * x * x * \dots * x \text{ (n个)}$
 $\text{val} = \text{val} * x; \text{ (val=1.0)}$

```
double power (double x, int n)
{
    double val = 1.0;

    while(n-->0)
        val = val * x;

    return(val);
}
```

```
double val=1.0;
```

```
n!=0
```

```
val=val*x;
```

```
n--;
```

```
return(val);
```

```

#include <iostream>
using namespace std;
void main( )
{
    double power (double x, int n);
    int a,b;
    double sum=0;
    cout<<"Input the a and b:";
    cin>>a>>b;
    sum=power((double)a,b)+power((double)b,a);
    cout << endl<<"The sum is: "<<sum<< endl;
}

```

输入a,b

调用子函数
求a的b次幂
和b的a次幂

输出结果

```

double power (double x, int n)
{
    double val = 1.0;
    while (n--)
        val = val*x;
    return(val);
}

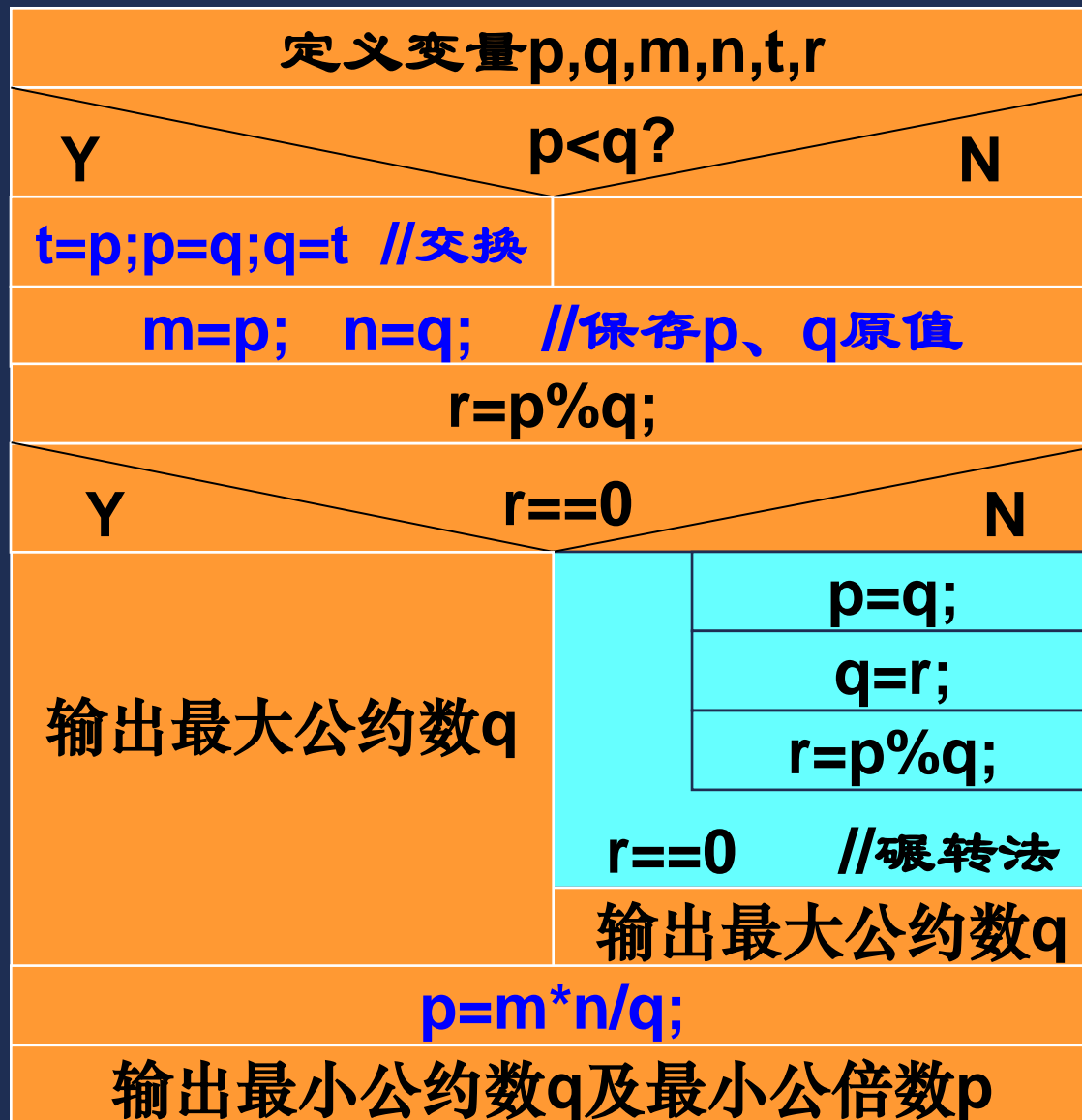
```

实验7-4

编写两个子函数，分别求两个整数的最大公约数和最小公倍数，并编写主函数调用这两个子函数，输出结果，两个整数由键盘输入。

- **分析：**假定两个整数分别为 p 和 q ，可用辗转法求最大公约数。
- **辗转法：**用 p 除以 q (保证 $p > q$)，求出余数 r ，如果 $r == 0$ ，则当前 q 就是最大公约数，如果 $r \neq 0$ ，令 $p = q$ ， $q = r$ ，重复以上过程，直到 $r == 0$ 为止。
- **最小公倍数 = 原来两数的积 / 最大公因数**

算法NS流程图



```
#include <iostream>
using namespace std;
void main( )
{
    int p, q, r,t,a,b; // 变量定义,p,q,r,a,b
    // 提示用户由键盘输入两个正整数
    cout<< "Please input two integer numbers:"<<endl;
    cin>>p>>q;
    a=p;
    b=q; //保留原来p,q的值, 分别放在a,b中
    //保证p为大数, q为小数
    if(p<q)
        { t=p;      p=q;      q=t;  }//交换p,q
```

如何设计子函数实现求最大公因数和最小公倍数？

int gys(int x, int y) {.....}

int gbs(int x, int y) {.....}

// 计算 p 除 q 的余数

r=p%q; // 只要

while(r!=0)

{ p=q;

q=r;

r=p%q;

} //辗转法

cout <<"两个正整数的最大公因数是"<<q<<endl;

cout <<"两个正整数的最小公倍数是"<<a*b/q<<endl;

}

int gbs(int x, int y)

{ return (x*y/gys(x,y)); }

算

int gys(int x, int y)

{

int r;

r=x%y;

while(r!=0)

{ x=y; y=r; r=x%y; }

return y;

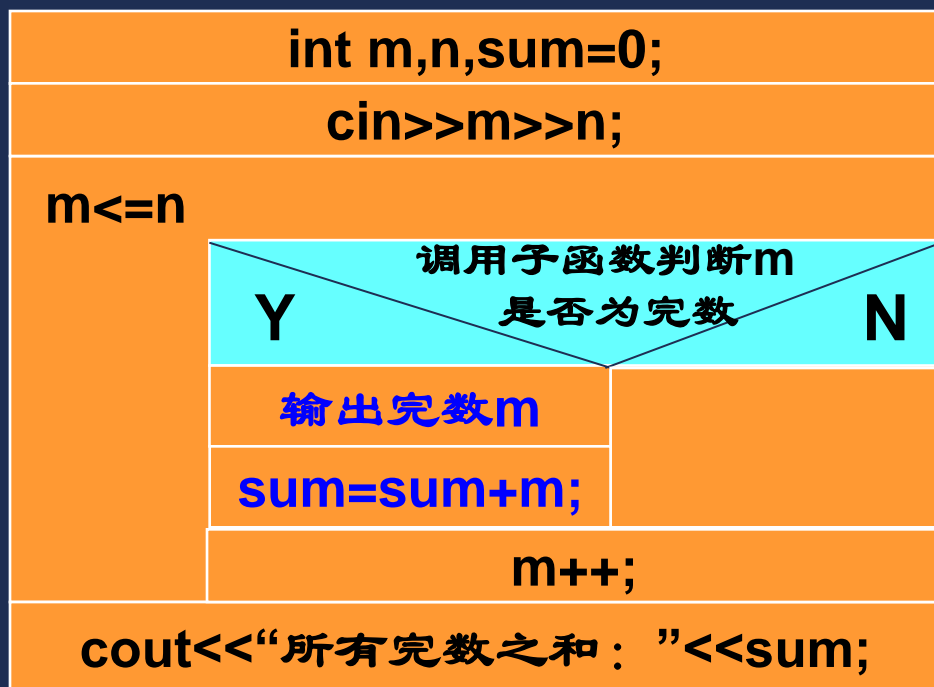
}

实验7-5

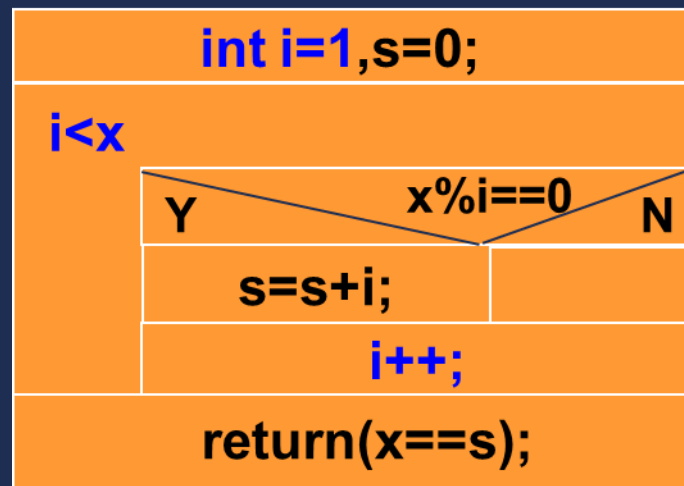
实现功能：计算正整数 $m \sim n$ 之间所有完数之和。

(1) **主函数功能**：从键盘输入两个整数 m 、 n ($m > 0, n > 0, m < n$)，通过调用子函数判断出 $m \sim n$ 之间所有完数，计算所有完数之和并输出结果。

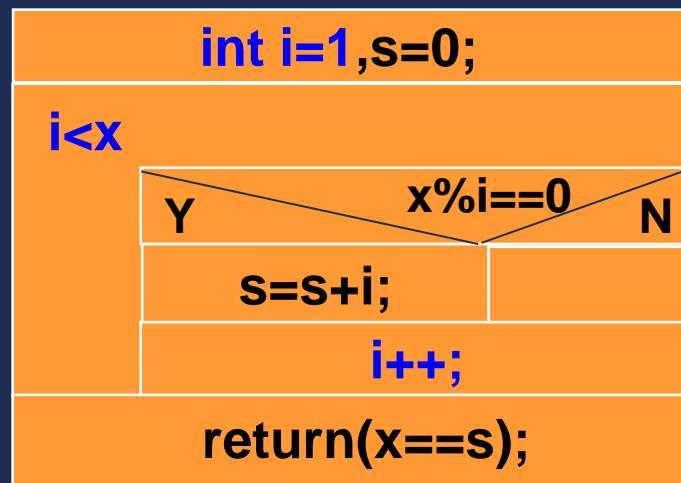
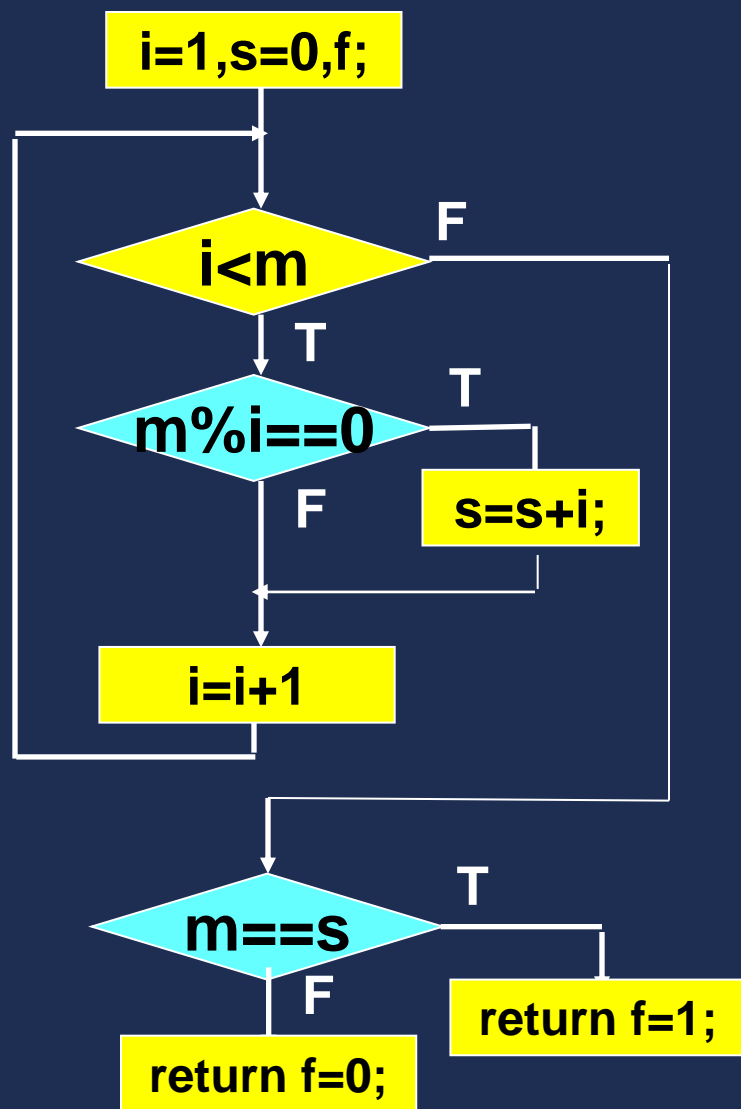
(2) **子函数功能**：定义有返回值函数，实现判断正整数是否为完数的功能。（**完数是指：该数所有因子（不含自身）之和等于该数，如 $28=1+2+4+7+14$** ）。



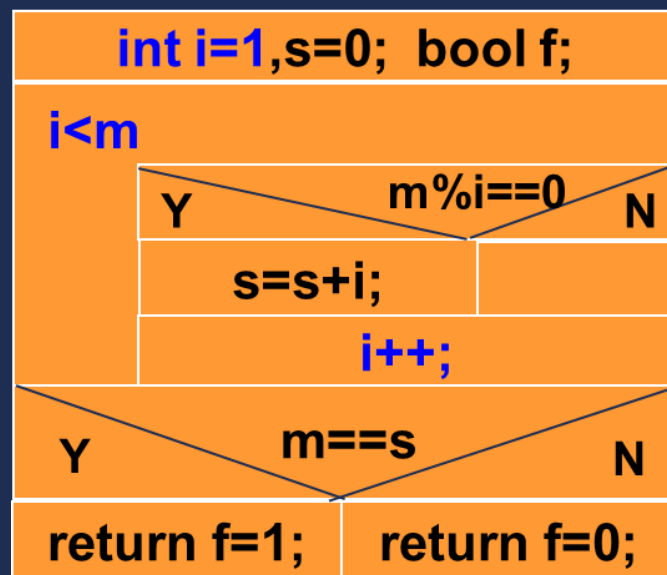
```
bool perfect_num(int x);
```



求完数的算法



求完数的算法



求完数的算法