从1到100相加：

public class Digui {

public int sum(int i){

if(i==1){

return 1;

}

return i+sum(i-1);

}

public static void main(String[] args) {

Digui test=new Digui();

System.out.println("计算结果："+test.sum(100)+"!");

}

}

从1到100阶乘：

需要注意的是计算后的结果数值过大程序无法返回，一般情况会返回0！那么用int、long 是无法满足的， 所以要用BigInteger

public class Digui {

public BigInteger sum(int i) {

if (i == 1) {

return BigInteger.ONE;

}

return BigInteger.valueOf(i).multiply(sum(i-1));

}

public static void main(String[] args) {

Digui test = new Digui();

try {

System.out.println("计算结果：" + test.sum(50) + "!");

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

题目：古典问题：3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

分析：首先我们要明白题目的意思指的是每个月的兔子总对数；假设将兔子分为小中大三种，兔子从出生后三个月后每个月就会生出一对兔子，

那么我们假定第一个月的兔子为小兔子，第二个月为中兔子，第三个月之后就为大兔子，那么第一个月分别有1、0、0，第二个月分别为0、1、0，

第三个月分别为1、0、1，第四个月分别为,1、1、1，第五个月分别为2、1、2，第六个月分别为3、2、3，第七个月分别为5、3、5……

兔子总数分别为：1、1、2、3、5、8、13……

于是得出了一个规律，从第三个月起，后面的兔子总数都等于前面两个月的兔子总数之和，即为斐波那契数列。

public class Test{

public static void main(String[] args){

int i = 1;

for(i=1;i<=20;i++){

System.out.println("兔子第"+i+"个月的总数为:"+f(i));

}

}

public static int f(int x){

if(x==1 || x==2){

return 1;

}else{

return f(x-1)+f(x-2);

}

}

}

图片包含 屏幕截图

描述已自动生成

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int N = input.nextInt(), Fn = 1, Fn1 = 0;//Fn来表示当前的斐波那契数值，Fn1用来表示Fn的上一个斐波那契数值

for (int i = 1; i < N; ++i) {//进行n-1次循环

int temp = (Fn + Fn1) % 10007;//temp为以Fn和Fn1为前两项的斐波那契数对10007的余数

Fn1 = Fn;

Fn = temp;//Fn和Fn1均指向下一个斐波那契数

}

System.out.println(Fn);

}

}

、图片包含 屏幕截图

描述已自动生成

import java.util.\*;

public class Main {

public static int gcd(int a,int b){//欧几里得算法求最大公约数

return b==0?a:gcd(b,a%b);

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int a=input.nextInt(),b=input.nextInt();

System.out.println(a/gcd(a,b)\*b);

}

}

图片包含 屏幕截图

描述已自动生成

import java.util.\*;

public class Main {

public static int f(int k,int n){

if(k==0||k==n)

return 1;

return f(k-1,n-1)+f(k,n-1);

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int a=in.nextInt(),b=in.nextInt();

System.out.println(f(a,b));

}

}

图片包含 屏幕截图

描述已自动生成

import java.util.\*;

public class Main {

public static int gcd(int a,int b){//欧几里得算法求解最大公约数

return b==0?a:gcd(b,a%b);

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int a=input.nextInt(),b=input.nextInt(),c=input.nextInt();

int k=a/gcd(a,b)\*b;//求出a,b的最小公倍数

System.out.println(k/gcd(k,c)\*c);//输出k,c的最小公倍数

}

}