**JAVA经典试题四十道**

程序1】 TestRabbit.java  
题目：古典问题：有一对兔子，从出生后第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？   
1.程序分析： 兔子的规律为数列1,1,2,3,5,8,13,21....

【程序2】 FindPrimeNumber.java  
题目：判断101-200之间有多少个素数，并输出所有素数。   
1.程序分析：判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除，   
则表明此数不是素数，反之是素数。

【程序3】FindDaffodilNumber.java  
题目：打印出所有的"水仙花数"，所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。例如：   
153是一个"水仙花数"，因为153=1的三次方＋5的三次方＋3的三次方。   
1.程序分析：利用for循环控制100-999个数，每个数分解出个位，十位，百位。

【程序4】FenJie.java  
题目：将一个正整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5。   
程序分析：对n进行分解质因数，应先找到一个最小的质数k，然后按下述步骤完成：   
(1)如果这个质数恰等于n，则说明分解质因数的过程已经结束，打印出即可。   
(2)如果n<>k，但n能被k整除，则应打印出k的值，并用n除以k的商,作为新的正整数你n,重复执行第一步。   
(3)如果n不能被k整除，则用k+1作为k的值,重复执行第一步。

【程序5】 ConditionOperator.java  
题目：利用条件运算符的嵌套来完成此题：学习成绩>=90分的同学用A表示，60-89分之间的用B表示，60分以下的用C表示。   
1.程序分析：(a>b)?a:b这是条件运算符的基本例子。

【程序6】Test1.java GcdTest.java后者是辗转相除法  
题目：输入两个正整数m和n，求其最大公约数和最小公倍数。   
1.程序分析：利用辗除法。

【程序7】 StChar.java  
题目：输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。   
1.程序分析：利用while语句,条件为输入的字符不为'\n'.

【程序8】 TestAdd.java  
题目：求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)，几个数相加有键盘控制。   
1.程序分析：关键是计算出每一项的值。

【程序9】 WanShu.java  
题目：一个数如果恰好等于它的因子之和，这个数就称为"完数"。例如6=1＋2＋3.编程 找出1000以内的所有完数。

【程序10】TestBall.java  
题目：一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第10次落地时，共经过多少米？第10次反弹多高？

【程序11】 TestTN.java

题目：有1、2、3、4个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？   
1.程序分析：可填在百位、十位、个位的数字都是1、2、3、4。组成所有的排列后再去 掉不满足条件的排列。

【程序12】 MoneyAward.java  
题目：企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万到60万之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%，高于100万元时，超过100万元的部分按1%提成，从键盘输入当月利润I，求应发放奖金总数？   
1.程序分析：请利用数轴来分界，定位。注意定义时需把奖金定义成长整型。

【程序13】FindNumber.java  
题目：一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，请问该数是多少？   
1.程序分析：在10万以内判断，先将该数加上100后再开方，再将该数加上268后再开方，如果开方后的结果满足如下条件，即是结果。请看具体分析：

【程序14】 TestDay.java  
题目：输入某年某月某日，判断这一天是这一年的第几天？   
1.程序分析：以3月5日为例，应该先把前两个月的加起来，然后再加上5天即本年的第几天，特殊情况，闰年且输入月份大于3时需考虑多加一天。

【程序15】TestCompare.java  
题目：输入三个整数x,y,z，请把这三个数由小到大输出。   
1.程序分析：我们想办法把最小的数放到x上，先将x与y进行比较，如果x>y则将x与y的值进行交换，然后再用x与z进行比较，如果x>z则将x与z的值进行交换，这样能使x最小。   
【程序16】Nine.java  
题目：输出9\*9口诀。   
1.程序分析：分行与列考虑，共9行9列，i控制行，j控制列。

【程序17】MonkeyEatPeach.java

题目：猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个 第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。   
1.程序分析：采取逆向思维的方法，从后往前推断。

【程序18】 Prog.java  
题目：两个乒乓球队进行比赛，各出三人。甲队为a,b,c三人，乙队为x,y,z三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a说他不和x比，c说他不和x,z比，请编程序找出三队赛手的名单。  
1.程序分析：判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除， 则表明此数不是素数，反之是素数。

【程序19】LingXing.java   
题目：打印出如下图案（菱形）   
    \*  
   \*\*\*  
 \*\*\*\*\*\*  
\*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*  
   \*\*\*  
    \*  
1.程序分析：先把图形分成两部分来看待，前四行一个规律，后三行一个规律，利用双重 for循环，第一层控制行，第二层控制列。

【程序20】TestAdd2.java  
题目：有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...求出这个数列的前20项之和。   
1.程序分析：请抓住分子与分母的变化规律。

【程序21】TestJieCheng.java  
题目：求1+2!+3!+...+20!的和   
1.程序分析：此程序只是把累加变成了累乘。

【程序22】   
题目：利用递归方法求5!。 TestJieCheng.java  
1.程序分析：递归公式：fn=fn\_1\*4!

【程序23】TestAge.java  
题目：有5个人坐在一起，问第五个人多少岁？他说比第4个人大2岁。问第4个人岁数，他说比第3个人大2岁。问第三个人，又说比第2人大两岁。问第2个人，说比第一个人大两岁。最后问第一个人，他说是10岁。请问第五个人多大？   
1.程序分析：利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人岁数，需知道第四人的岁数,依次类推，推到第一人（10岁），再往回推。

【程序24】TestNumber.java  
题目：给一个不多于5位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。

【程序25】 HuiWenShu.java  
题目：一个5位数，判断它是不是回文数。即12321是回文数，个位与万位相同，十位与千位相同。   
【程序26】Ex26.java  
题目：请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续 判断第二个字母。   
1.程序分析：用情况语句比较好，如果第一个字母一样，则判断用情况语句或if语句判断第二个字母。   
【程序27】 SuShu.java  
题目：求100之内的素数   
【程序28】 TestSort.java  
题目：对10个数进行排序   
1.程序分析：可以利用选择法，即从后9个比较过程中，选择一个最小的与第一个元素交换， 下次类推，即用第  
  
二个元素与后8个进行比较，并进行交换。   
【程序29】 TestAdd3.java  
题目：求一个3\*3矩阵对角线元素之和   
1.程序分析：利用双重for循环控制输入二维数组，再将a[i][i]累加后输出。   
【程序30】 ArraySort.java  
题目：有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中。   
1. 程序分析：首先判断此数是否大于最后一个数，然后再考虑插入中间的数的情况，插入后此元素之后的数，   
  
依次后移一个位置。   
【程序31】 ArrayConverse.java  
题目：将一个数组逆序输出。   
1.程序分析：用第一个与最后一个交换。   
【程序32】 Ex32.java  
题目：取一个整数a从右端开始的4～7位。   
程序分析：可以这样考虑：   
(1)先使a右移4位。   
(2)设置一个低4位全为1,其余全为0的数。可用~(~0<<4)  
(3)将上面二者进行&运算。   
【程序33】YangHui.java  
题目：打印出杨辉三角形（要求打印出10行如下图）   
1.程序分析：   
     1  
    1 1  
   1 2 1  
  1 3 3 1  
 1  4 6 4 1  
1 5 10 10 5 1  
  
【程序34】 略 前面更复杂的已经做过了  
题目：输入3个数a,b,c，按大小顺序输出。   
1.程序分析：利用指针方法。   
【程序35】 ArrayChange.java  
题目：输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。   
【程序36】 Array1.java  
题目：有n个整数，使其前面各数顺序向后移m个位置，最后m个数变成最前面的m个数   
【程序37】 Test3Quit.java  
题目：有n个人围成一圈，顺序排号。从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的那位。   
【程序38】 TestLength.java  
题目：写一个函数，求一个字符串的长度，在main函数中输入字符串，并输出其长度。   
【程序39】 Test2.java  
题目：编写一个函数，输入n为偶数时，调用函数求1/2+1/4+...+1/n,当输入n为奇数时，调用函数   
  
1/1+1/3+...+1/n(利用指针函数)  
【程序40】 Test3.java  
题目：字符串排序。   
【程序41】 MonkeyPeach.java  
题目：海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子凭据分为五份，多了一个，这只猴子把多的一   
  
个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，又多了一个，它同样把多的一个扔入海中   
  
，拿走了一份，第三、第四、第五只猴子都是这样做的，问海滩上原来最少有多少个桃子？   
  
【程序42】 Test4.java  
题目：809\*??=800\*??+9\*??+1  
其中??代表的两位数,8\*??的结果为两位数，9\*??的结果为3位数。求??代表的两位数，及809\*??后的结果。   
【程序43】 Test5.java  
题目：求0—7所能组成的奇数个数。   
【程序44】 TestEven.java  
题目：一个偶数总能表示为两个素数之和。   
【程序45】TestPrime9.java  
题目：判断一个素数能被几个9整除   
【程序46】 TestString.java  
题目：两个字符串连接程序   
【程序47】 TestPrint.java  
题目：读取7个数（1—50）的整数值，每读取一个值，程序打印出该值个数的＊。   
【程序48】 TestCode.java  
题目：某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：每位数字   
  
都加上5,然后用和除以10的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换。   
  
【程序49】 TestString2.java  
题目：计算字符串中子串出现的次数   
【程序50】TestStu.java

题目：有五个学生，每个学生有3门课的成绩，从键盘输入以上数据（包括学生号，姓名，三门课成绩），计算

。。。。。。。。。。。我是分割线。。。。。。。。。。。。。

原创代码，需要分析的朋友可以留言，小白会第一时间认真解答，要是有哪里不对的请及时帮我指出。

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/28.
5. \*/
6. /\*\*
7. \* 有5个人坐在一起，
8. \* 问第五个人多少岁？他说比第4个人大2岁。
9. \* 问第4个人岁数，他说比第3个人大2岁。
10. \* 问第三个人，又说比第2人大两岁。
11. \* 问第2个人，说比第一个人大两岁。
12. \* 最后问第一个人，他说是10岁。
13. \* 请问第五个人多大？
14. \*
15. \*
16. \*  使用递归算法
17. \*
18. \* \*/
19. **public** **class** Age {
20. **public** **static** **void** main(String[] args){
21. System.out.print(getAge(5));
22. }
23. **public** **static** **int** getAge(**int** person){
24. **if**(person == 1)
25. **return** 10;
26. **else**
27. **return** getAge(person - 1) + 2;
28. }
29. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. **import** java.util.Scanner;
5. /\*\*
6. \* Created by sakura on 16/5/30.
7. \*/
8. /\*\*
9. \*
10. \*       将一个数组逆序输出
11. \*
12. \*      当a数组以实参传入函数Fun()时,函数内部对形参的操作改变了原始数组a的值,
13. \*      这是因为数组传递的是引用,对引用的操作就是对原本数组的操作
14. \*
15. \* \*\*/
16. **public** **class** ArrayChange {
17. **public** **static** **void** main(String[] args){
18. Scanner in = **new** Scanner(System.in);
19. **int** a[] = **new** **int**[5];
20. **for**(**int** fi = 0;fi < a.length;fi++){
21. a[fi] = in.nextInt();
22. }
23. Fun(a);
24. **for**(**int** fj = 0;fj < a.length;fj++){
25. System.out.print(a[fj]);
26. }
27. }
28. **public** **static** **void** Fun(**int**[] a){
29. **int** i,j,k;
30. **int** time;
31. **if**(a.length % 2 == 0){
32. i = 0;
33. j = a.length - 1;
34. **while**(i + 1 != j){
35. time = a[i]; a[i] = a[j]; a[j] = time;
36. i++; j--;
37. }
38. }**else**{
39. i = 0;
40. j = a.length - 1;
41. k = (a.length + 1) / 2 - 1;     //中间数组的下标
42. **while**(i != k){
43. time = a[i]; a[i] = a[j]; a[j] = time;
44. i++; j--;
45. }
46. }
47. }
48. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/14.
5. \*/
6. /\*\*
7. \* 一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第10次落地时，共经过多少米？第10次反弹多高？
8. \*
9. \* \*/
10. **public** **class** Ball {
11. **public** **static** **void** main(String[] args){
12. **int** time = 1;
13. **double** h = 100;
14. **double** totalH = 0;
15. **while**(time <= 10){
16. totalH = totalH + h + h/2;
17. h = h/2;
18. time++;
19. }
20. System.out.println("10次落地共经过: "+totalH+"\n第10次反弹高度为 "+h);
21. }
22. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
2. **import** **static** java.lang.Math.pow;
3. /\*\*
4. \* Created by sakura on 16/5/12.
5. \*/
6. /\*\*
7. \* 题目：求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+222,2+222,22(此时共有5个数相加)，
8. \* 几个数相加有键盘控制。
9. \*  1.程序分析：关键是计算出每一项的值。
10. \*
11. \*          答案正确
12. \* \*/
13. **public** **class** composition {
14. **public** **static** **void** main(String[] args){
15. System.out.println(calculate(1,10));
16. }
17. **public** **static** **int** calculate(**int** baseData,**int** comTime){
18. **int** total = 0;
19. **int** time=0;
20. **int** nowData = 0;
21. **int** lastData = 0;
22. **while**(time < comTime){
23. nowData = baseData \* (**int**)pow(10,time++) + lastData;
24. total = total + nowData;
25. lastData = nowData;
26. System.out.println(nowData);
27. }
28. **return** total;
29. }
30. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/10.
5. \*/
6. /\*\*
7. \* 将一个正整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5
8. \*
9. \*          已测试 答案正确
10. \* \*/
12. **import** java.util.Scanner;
13. **import** java.lang.\*;
14. **public** **class** DecomposeInteger {
15. **public** **static** **void** main(String[] args){
16. **int**[] a = **new** **int**[30];
17. **int** i = 0;
18. **int** number = 0;
19. **int** time = 2;
20. Scanner in =**new** Scanner(System.in);
21. **try**{
22. number = in.nextInt();
23. }**catch**(Exception e){
24. System.out.println("输入有误!");
25. }
26. **while**(calculatePrimeNumber(number) == **false**) {
27. **while** (number % time != 0) {
28. time++;
29. }
30. number = number / time;
31. a[i++] = time;
32. time = 2;
33. }
34. a[i] = number;
35. **for**(**int** forTime = 0;forTime <= i;forTime++){
36. **if**(forTime == 0){
37. System.out.print("="+a[forTime]);
38. }**else** {
39. System.out.print("\*"+a[forTime]);
40. }
41. }
43. }
44. **public** **static** **boolean** calculatePrimeNumber(**int** number){
45. **for**(**int** forTime = 2;forTime <=Math.sqrt(number);forTime++){
46. **if**(number%forTime == 0){
47. **return** **false**;
48. }
49. }
50. **return** **true**;
51. }
52. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/15.
5. \*/
6. /\*\*
7. \* 求1+2!+3!+...+20!的和
8. \*
9. \* !是阶乘的符号。对于所有正整数n，n!=1\*2\*3... \*n；而规定0!=1
10. \*
11. \*              答案正确 268040729
12. \*
13. \* \*/
14. **public** **class** Factorial {
15. **public** **static** **void** main(String[] args){
16. **int** count = 20;
17. **int** i = 1;
18. **int** total = 0;
19. **while**(i <= count){
20. total += calculate(i++);
21. }
22. System.out.println(total);
23. }
24. **public** **static** **int** calculate(**int** number){
25. **int** i = 1;
26. **int** total = 1;
27. **while**(i <= number){
28. total = total \* i;
29. i++;
30. }
31. **return** total;
32. }
33. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/15.
5. \*/
6. //    \*
7. //   \*\*\*
8. //  \*\*\*\*\*
9. // \*\*\*\*\*\*\*
10. //  \*\*\*\*\*
11. //   \*\*\*
12. //    \*
13. **public** **class** Figure {
14. **public** **static** **void** main(String[] args){
15. **int** rbombusH = 10;
16. **int** space = (rbombusH - 1) / 2 ;
17. **int** countUp = 1;
18. **int** whileTime = 1;
19. **int** i = 0;
20. **while**(whileTime <= (rbombusH - 1) / 2){
21. i = 0;
22. **while**(i < space) {
23. System.out.print(" ");
24. i++;
25. }
26. space -= 1;
27. i = 0;
28. **if**(whileTime == 1)
29. **while**(i < countUp) {
30. System.out.print("\*");
31. i++;
32. }
33. **else**
34. **while**(i < countUp) {
35. System.out.print("\*");
36. i++;
37. }
38. System.out.print("\n");
39. countUp += 2;
40. whileTime++;
41. }
43. i = 0;
44. **while**(i < countUp){
45. System.out.print("\*");
46. i++;
47. }
48. System.out.print("\n");
49. whileTime = 1;
50. space = 1;
51. **while**(whileTime <= (rbombusH - 1) / 2){
52. countUp -= 2;
53. i = 0;
54. **while**(i < space){
55. System.out.print(" ");
56. i++;
57. }
58. i = 0;
59. **while**(i < countUp){
60. System.out.print("\*");
61. i++;
62. }
63. System.out.print("\n");
64. whileTime++;
65. space++;
66. }
67. }
68. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/28.
5. \*/
7. **import** java.util.Scanner;
9. /\*\*
10. \*有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中
11. \*
12. \*      //SP 11112345的情况 开头几个数字是相同的情况
13. \* \*/
14. **public** **class** InsertNumberToArray {
15. **public** **static** **int** arrayCount = 6;   //设置有几个数字
16. **public** **static** **void** main(String[] args){
17. Scanner in = **new** Scanner(System.in);
18. **int**[] number = **new** **int**[100];
19. **for**(**int** i = 0;i < arrayCount;i++){
20. number[i] = in.nextInt();
21. }
22. **int**[] newNumber = Fun(number,9);
23. **for**(**int** i = 0;i < arrayCount + 1;i++){
24. System.out.println(newNumber[i]);
25. }
26. }
27. **public** **static** **int**[] Fun(**int**[] number,**int** i){
28. **if**(number[0] > number[1]){
29. **if**(i >= number[0]){
30. **for**(**int** time = arrayCount;time > 0;time--){
31. number[time] = number[time - 1];
32. }
33. number[0] = i ;
34. }**else**{
35. **int** time = 0;
36. **while**(**true**){
37. **if**(number[time] >= i && number[time + 1] <= i){
38. **break**;
39. }
40. **if**(time != arrayCount - 1)
41. time++;
42. **else**
43. **break**;
44. }
45. time++;
46. **for**(**int** time1 = arrayCount - 1;time1 >= time;time1 --){
47. number[time1 + 1] = number[time1];
48. }
49. number[time] = i;
50. }
51. **return** number;
52. }**else** **if**(number[0] < number[1]){
53. **if**(i <= number[0]){
54. **for**(**int** time = arrayCount;time > 0;time--){
55. number[time] = number[time - 1];
56. }
57. number[0] = i ;
58. }**else**{
59. **int** time = 0;
60. **while**(**true**){
61. **if**(number[time] <= i && number[time + 1] >= i){
62. **break**;
63. }
64. **if**(time != arrayCount - 1)
65. time++;
66. **else**
67. **break**;
68. }
69. time++;
70. **for**(**int** time1 = arrayCount - 1;time1 >= time;time1 --){
71. number[time1 + 1] = number[time1];
72. }
73. number[time] = i;
74. }
75. **return** number;
76. }**else**{
77. **int** k = 1;
78. **int** j = 2;
79. **int** count = 0;
80. **int** newArrayCount;
81. **while**(number[k] == number[j]){
82. k++;
83. j++;
84. }
85. newArrayCount = arrayCount - k - 1;   //除了开头相同的数字剩下有几个数字
86. **if**(number[k] > number[j]){
87. **if**(i >= number[0]){
88. **for**(**int** time = arrayCount;time > 0;time--){
89. number[time] = number[time - 1];
90. }
91. number[0] = i ;
92. }**else**{
93. **int** time = k;
94. **while**(**true**){
95. **if**(number[time] >= i && number[time + 1] <= i){
96. **break**;
97. }
98. **if**(time != arrayCount - 1)
99. time++;
100. **else**
101. **break**;
102. }
103. time++;
104. **for**(**int** time1 = arrayCount - 1;time1 >= time;time1 --){
105. number[time1 + 1] = number[time1];
106. }
107. number[time] = i;
108. }
109. **return** number;
110. }**else** **if**(number[k] < number[j]){
111. **if**(i <= number[k]){
112. **for**(**int** time = arrayCount;time > 0;time--){
113. number[time] = number[time - 1];
114. }
115. number[0] = i ;
116. }**else**{
117. **int** time = k;
118. **while**(**true**){
119. **if**(number[time] <= i && number[time + 1] >= i){
120. **break**;
121. }
122. **if**(time != arrayCount - 1)
123. time++;
124. **else**
125. **break**;
126. }
127. time++;
128. **for**(**int** time1 = arrayCount - 1;time1 >= time;time1 --){
129. number[time1 + 1] = number[time1];
130. }
131. number[time] = i;
132. }
133. **return** number;
134. }
135. }
136. **return** number;
137. }
138. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/14.
5. \*/
6. /\*\*\*
7. \*
8. \*一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，请问该数是多少
9. \*/
10. **public** **class** JudgeNumber {
11. **public** **static** **void** main(String[] args){
12. **int** i = 11;
13. **int** number =9;
14. //System.out.println((int)Math.sqrt(5));
15. **while**(**true**){
16. **if**(calculate(number + 100) == **true** && calculate(number + 268) == **true**){
17. System.out.println(number);
18. **break**;
19. }
20. number++;
21. }
22. }
23. **public** **static** **boolean** calculate(**int** number){
24. **if**((**int**)Math.sqrt(number) \* (**int**)Math.sqrt(number) == number)
25. **return** **true**;
26. **return** **false**;
27. }
28. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/14.
5. \*/
7. **import** java.util.Scanner;
9. /\*\*
10. \* 企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；
11. \* 利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可可提成7.5%；
12. \* 20万到40万之间时，高于20万元的部分，可提成5%；
13. \* 40万到60万之间时高于40万元的部分，可提成3%；
14. \* 60万到100万之间时，高于60万元的部分，可提成1.5%，
15. \* 高于100万元时，超过100万元的部分按1%提成，
16. \* 从键盘输入当月利润I，求应发放奖金总数？
17. \*
18. \*
19. \* \*/
20. **public** **class** MoneyAward {
21. **public** **static** **void** main(String[] args){
22. Scanner in = **new** Scanner(System.in);
23. **double** profit = in.nextDouble();
24. **double** moneyAward = 0;
25. **if**(profit <= 100000){
26. moneyAward = profit \* 0.1;
27. }**else** **if**(profit < 200000){
28. moneyAward = 100000 \* 0.1 + (profit - 100000) \* 0.075;
29. }**else** **if**(profit < 400000){
30. moneyAward = 200000 \* 0.1 + (profit - 200000) \* 0.05;
31. }**else** **if**(profit < 600000){
32. moneyAward = 400000 \* 0.1 +(profit - 400000) \* 0.03;
33. }**else** **if**(profit < 1000000){
34. moneyAward = 600000 \* 0.1 +(profit - 600000) \* 0.15;
35. }**else**{
36. moneyAward = 1000000 \* 0.1 +(profit - 1000000) \* 0.01;
37. }
38. System.out.println(moneyAward);
39. }
40. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/15.
5. \*/
6. /\*\*
7. \* 猴子吃桃问题：
8. \* 猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个
9. \* 第二天早上又将剩下的桃子吃掉一半，又多吃了一个。
10. \* 以后每天早上都吃了前一天剩下的一半零一个。
11. \* 到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。
12. \*
13. \*      答案: 1534个
14. \*
15. \* \*/

18. **public** **class** MonkeyEat {
19. **public** **static** **void** main(String[] args){
20. **int** peach = 1;
21. **int** whileTime = 9;
22. **while**(whileTime >= 1){
23. peach = (peach + 1) \* 2;
24. whileTime--;
25. }
26. System.out.println(peach);
27. }
28. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/6/6.
5. \*/
6. /\*\*
7. \*  题目：海滩上有一堆桃子，五只猴子来分。
8. \*  第一只猴子把这堆桃子凭据分为五份，多了一个，
9. \*  这只猴子把多的一   个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，
10. \*  又多了一个，它同样把多的一个扔入海中   ，拿走了一份，第三、第四、第五只猴子都是这样做的，
11. \*  问海滩上原来最少有多少个桃子？
12. \*
13. \* \*/
14. **public** **class** MonkeySeabeach {
15. **public** **static** **void** main(String[] args){
16. **int** x = 2;
17. **int** m = 1;
18. **int** y = 0;
19. **int** i = 0;
20. **int** t = 0;
21. **while**(**true**){
22. **for**(i = 0 ; i < 5 ; i ++){
23. t = 0;
24. **if**(i == 0)
25. y = x;
26. **if**((x - 1) % 5 == 0){
27. x = (x - 1) - (x - 1) / 5;
28. }**else**{
29. t = 1;
30. x = y;
31. **break**;
32. }
33. }
34. **if**(i == 5 && t == 0)            //i == 5 ,  不是 i == 4 !!!!!!!!
35. **break**;
36. x++;
37. }
38. System.out.println("y="+y);
39. }
40. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
2. **import** **static** java.lang.Math.pow;
3. /\*\*
4. \* Created by sakura on 16/5/10.
5. \*/
6. /\*\*
7. \* 【程序3题目：打印出所有的"水仙花数"，所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。
8. \* 例如：  153是一个"水仙花数"，因为153=1的三次方＋5的三次方＋3的三次方。
9. \*
10. \*      答案正确
11. \* \*/
12. **public** **class** Narcissus {
13. **public** **static** **void** main(String[] args){
14. **int** number = 100;
15. **int** count = 0;
16. **while**(number <= 999){
17. **if**(calculateNarcissus(number)){
18. System.out.println(number);
19. count++;
20. }
21. number++;
22. }
23. System.out.println("共有"+count+"个");
24. }
25. **public** **static** **boolean** calculateNarcissus(**int** number){
26. **if**(number == pow((number%10),3) + pow((number%100 - number%10)/10,3) + pow((number - number%100)/100,3))
27. **return** **true**;
28. **return** **false**;
29. }
30. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/10.
5. \*/
6. /\*\*
7. \* 题目：利用条件运算符的嵌套来完成此题：学习成绩>=90分的同学用A表示，60-89分之间的用B表示，60分以下的用C表示。
8. \*  1.程序分析：(a>b)?a:b这是条件运算符的基本例子。
9. \*
10. \*     答案正确
11. \* \*/
12. **public** **class** Nesting {
13. **public** **static** **void** main(String[] args){
14. **int** grade = 61;
15. System.out.println((grade>=90)?"A":(grade>=60)?"B":"C");
16. }
17. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/31.
5. \*/
7. **import** java.util.Scanner;
9. /\*\*
10. \*      输入数组，最大的与第一个元素交换，最小的与最后一个
11. \*
12. \* \*/
13. **public** **class** NumberChange {
14. **public** **static** **int** numberCount = 5;
15. **public** **static** **void** main(String[] args){
16. Scanner in = **new** Scanner(System.in);
17. **int** a[] = **new** **int**[100];
18. **int** i;
19. **for**(i = 0; i < numberCount; i++){
20. a[i] = in.nextInt();
21. }
22. **int** m = a[0],n = 0,time;
23. **int** b[] = a.clone();
24. i = 0;
25. **while**(a[i] != '\0'){
26. **if**(a[i + 1] != '\0') {
27. **if** (a[i + 1] >= m){
28. m = a[i + 1];
29. n = i + 1;
30. }
31. }
32. i++;
33. }
34. time = a[0];
35. a[0]= m;
36. a[n] = time;
37. i = 0;
38. **while**(a[i] != '\0'){
39. System.out.print(a[i]+" ");
40. i++;
41. }
42. System.out.println();
44. i = 0;
45. m = b[0];
46. n = 0;
47. **while**(b[i] != '\0'){
48. **if**(b[i + 1] != '\0'){
49. **if**(b[i + 1] < m){
50. m = b[i + 1];
51. n = i + 1;
52. }
53. }
54. i++;
55. }
56. time = b[numberCount - 1];
57. b[numberCount - 1] = m;
58. b[n] = time;
59. i = 0;
60. **while**(b[i] != '\0'){
61. System.out.print(b[i]+" ");
62. i++;
63. }
65. }
66. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
2. //import static java.lang.Math.sqrt;        //如果用这种方式导入 调用sqr方法是不需要使用Math.来调用 可直接使用函数sqrt()
3. **import** java.lang.\*;
4. /\*\*
5. \* Created by sakura on 16/5/10.
6. \*/
7. /\*\*
8. \*
9. \*  题目：判断101-200之间有多少个素数，并输出所有素数。
10. 1.程序分析：判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除，
11. 则表明此数不是素数，反之是素数。
13. 已测试,答案正确
15. \*/
16. **public** **class** PrimeNumber {
17. **public** **static** **void** main(String [] args){
18. **int** number = 101,count = 0;
19. **while**(number <= 200){
20. **if**(calculatePrimeNumber(number) == **false**){
21. count++;
22. System.out.println(number+"是素数");
23. }
24. number++;
25. }
26. System.out.println("共有"+count+"个素数");
27. }
28. **public** **static** **boolean** calculatePrimeNumber(**int** number){
29. **for**(**int** forTime = 2;forTime <=Math.sqrt(number);forTime++){
30. **if**(number%forTime == 0){
31. **return** **true**;
32. }
33. }
34. **return** **false**;
35. }
37. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/9.
5. \*/
6. /\*\*
7. 题目：古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，
8. 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？
9. 1.程序分析： 兔子的规律为数列1,1,2,3,5,8,13,21....
11. 解:每一个月兔子总数等于前两个月的兔子总数之和
12. 已测试 答案正确
13. \*\*/
14. **public** **class** Rabbit {
15. **public** **static** **void** main(String [] args){
16. **int**[] rabbit = **new** **int**[100];
17. **int** month = 0;
18. rabbit[0] = 1;
19. **while**(month <= 23){
20. **if**(month == 0){
21. System.out.printf("第%d月 = %d\n",month+1,rabbit[month]);
22. }
23. **if**(month == 1){
24. rabbit[month] = rabbit[month-1];
25. System.out.printf("第%d月 = %d\n",month+1,rabbit[month]);
26. }
27. **if**(month > 1) {
28. rabbit[month] = rabbit[month - 1] + rabbit[month - 2];
29. System.out.printf("第%d月 = %d\n", month + 1, rabbit[month]);
30. }
31. month++;
32. }
34. }
35. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/28.
5. \*/
6. /\*\*
7. \*  利用递归方法求5!
8. \*
9. \* 网友答案:
10. \*   public int getResult(int n){
11. if(n==1){
12. return 1;
13. }else{
14. return n\*getResult(n-1);
15. }

18. \* \*/
19. **public** **class** RecurrenceJieCheng {
20. **public** **static** **final** **int** time = 4;
21. **public** **static** **int** total = 1;
23. **public** **static** **void** main(String[] args) {
24. System.out.println(JieCheng2(1));
25. //System.out.println(getResult(4));

28. }
30. **public** **static** **int** JieCheng(**int** a) {  //不是完美递归
31. **if** (a < time) {
32. total += total \* a;
33. a++;
34. JieCheng(a);
35. }
36. **return** total;
37. }
38. **public** **static** **int** JieCheng2(**int** a){     //是递归,不过递归防方向不同,使用了final数据time,使程序繁琐
39. **if**(a > time)
40. **return** 1;
41. **else**{
42. **return** a \* JieCheng2(a + 1);
43. }
45. }
46. **public** **static** **int** getResult(**int** n) {   //完美递归,来自百度知道
47. **if** (n == 1) {
48. **return** 1;
49. } **else** {
50. **return** n \* getResult(n - 1);
51. }
52. }
53. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/12.
5. \*/
7. /\*\*
8. \* 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数
9. \*          答案正确
10. \*
11. \* \*/
12. **public** **class** StatisticsString {
13. **public** **static** **void** main(String[] args){
14. **int** countGrapheme = 0,countDigit = 0,countSpace = 0,countOther = 0;
15. String string =**new** String("askqiwu       1123109q2 w\*\*\*&^%$#$%1 3m ");
16. **int** time = 0;
17. **while**(time < string.length()){      //如果加了一个= 变成 time <= string.length() 就会报错,越界
18. **if**(((**int**)string.charAt(time) >= 65 && (**int**)string.charAt(time) <= 90) || ((**int**)string.charAt(time) >= 97 && (**int**)string.charAt(time) <= 122))
19. countGrapheme++;
20. **else** **if**((**int**)string.charAt(time) >= 48 && (**int**)string.charAt(time) <= 57)
21. countDigit++;
22. **else** **if**((**int**)string.charAt(time) == 32)
23. countSpace++;
24. **else**
25. countOther++;
27. time++;
28. }
29. System.out.println("Grapheme = "+countGrapheme+"\n"+"Digit = "+countDigit+"\n"+"Space = "+countSpace+"\n"+"Other"+countOther);
30. }
31. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/6/3.
5. \*/
6. /\*\*
7. \*  写一个函数，求一个字符串的长度，在main函数中输入字符串，并输出其长度。
8. \*
9. \* \*/
10. **public** **class** StringLengh {
11. **public** **static** **void** main(String[] args) {
12. System.out.println(Fun("absdn hsjnah"));
13. }
15. **public** **static** **int** Fun(String string) {
16. **int** i = 0;
17. **try** {
18. **while** (string.charAt(i) != '\0')
19. i++;
20. } **catch** (Exception e) {
21. **return** i;
22. }
23. **return** 0;
24. }
25. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. **import** java.util.Scanner;
5. /\*\*
6. \* Created by sakura on 16/6/5.
7. \*/
8. /\*\*
9. \*  字符串排序
10. \*
11. \*  \*/
12. **public** **class** StringRank {
13. **public** **static** **void** main(String[] args){
14. Scanner in  = **new** Scanner(System.in);
15. **int** number = 4;
16. String[] string = **new** String[20];
17. **for**(**int** forTime = 0;forTime < number;forTime ++) {
18. string[forTime] = in.next();
19. }
20. **int** i ,j,k;
21. **int** time = 0;
22. String change = **new** String();
23. change = **null**;
24. **for**(i = 0; i < number ; i ++){
25. **for**(j = i + 1; j < number; j++){
26. k = 0;
27. time = 0;
28. **while**(time == 0 && k < string[i].length() && k < string[j].length()){
29. **if**(string[i].charAt(k) > string[j].charAt(k)){
30. time = 1;
31. }**else** **if**(string[i].charAt(k) < string[j].charAt(k)){
32. change = string[i]; string[i] = string[j]; string[j] = change;
33. time = 1;
34. }**else** **if**(string[i].charAt(k) == string[j].charAt(k)) {
35. **if** (k == string[i].length() - 1 && k != string[j].length() - 1) {   //abc abcd的情况
36. change = string[i];
37. string[i] = string[j];
38. string[j] = change;
39. time = 1;
40. } **else** {
41. time = 0;
42. k++;
43. }
44. }
45. }
46. }
47. }
48. **for**(**int** forTime = 0; forTime < number ; forTime ++){
49. System.out.println(string[forTime]);
50. }
51. }
52. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/14.
5. \*/
6. /\*\*
7. \*
8. \* 有1、2、3、4个数字，能组成多少个 互不相同 且 无重复数字 的三位数？都是多少？
9. \*
10. \*      利用for!!!!!no-complicatino
11. \*
12. \* \*/
13. **public** **class** Test1 {
14. **public** **static** **void** main(String[] args){
15. **int** i,j,k,count = 0;
16. **for**(i = 1;i <= 4;i++){
17. **for**(j = 1;j <= 4;j++){
18. **if**(i != j){
19. **for**(k = 1;k <= 4;k++){
20. **if**(k!=i && k!=j){
21. System.out.println(i\*100+j\*10+k);
22. count++;
23. }
24. }
25. }
26. }
27. }
28. System.out.println("共有"+count);
29. }
30. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/15.
5. \*/
6. /\*\*
7. \* 有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...求出这个数列的前20项之和。
8. \*
9. \*
10. \*                 答案正确
11. \* \*/
12. **public** **class** Test2 {
13. **public** **static** **void** main(String[] args){
14. **int** count = 20;
15. **double** up = 2, down = 1;
16. **double** total = 0;
17. **int** i = 1;
18. **double** time = 0;
19. **while**(i <= count){
20. total += up / down;
21. time = down;
22. down = up;
23. up = time + up;
24. i++;
25. }
26. System.out.println(total);
27. }
28. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/28.
5. \*/
7. /\*\*
8. \*
9. \* \*   给一个正整数，要求：一、求它是几位数，二、逆序打印出各位数字。
10. \* \*\*/
11. **public** **class** Test3 {
12. **public** **static** **void** main(String[] atgs){
13. System.out.print(getNumber("1989898980"));
14. }
15. **public** **static** **int** getNumber(String string){
16. **int** oriNumber = Integer.valueOf(string);
17. **int** newNumber = 0;
18. **int** numberCount = string.length();
19. **int** time = string.length() - 1;
20. **for**(**int** i = 1;i <= string.length();i++){
21. newNumber += getSpNumber(i,oriNumber) \* Math.pow(10,time--);
22. }
23. **return** newNumber;
24. }
25. **public** **static** **int** getSpNumber(**int** a,**int** number){
26. **if**(a == 1){
27. **return** number % 10;
28. }**else**{
29. **return** (**int**) ((**int**) ((number % Math.pow(10,a)) - (number % Math.pow(10,a-1))) / Math.pow(10,(a - 1)));
30. }
31. }
32. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. **import** java.util.Scanner;
5. /\*\*
6. \* Created by sakura on 16/5/30.
7. \*/
8. /\*\*
9. \*
10. \*      取一个整数a从右端开始的4～7位
11. \*
12. \* \*/
13. **public** **class** Test4 {
14. **public** **static** **void** main(String[] atgs){
15. Scanner in = **new** Scanner(System.in);
16. **int** a = in.nextInt();
17. **int** n = ( a % (**int**) Math.pow(10,7) - a % (**int**) Math.pow(10,3) ) / 1000;
18. System.out.println(n);
19. }
20. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/6/1.
5. \*/
7. **import** java.util.Scanner;
9. /\*\*
10. \*      有n个整数，使其前面各数顺序向后移m个位置，最后m个数变成最前面的m个数
11. \*
12. \* \*/
13. **public** **class** Test5 {
14. **public** **static** **void** main(String[] args){
15. Scanner in = **new** Scanner(System.in);
16. **int** a[] = **new** **int**[100];
17. **int** count = 10;
18. **int** m= 9;
19. **for**(**int** i =0; i < count ; i ++){
20. a[i] = in.nextInt();
21. }
22. **int** k = count - m;
23. **int** timeCount = count;
24. **for**(**int** j = count - m; j < count; j ++){
25. a[timeCount++] = a[j];
26. }
27. System.out.println("\*\*" + a[6]+"\*\*"+a[7]);
28. k = count - 1;
29. **for**(**int** j = count - m - 1; j >= 0; j--){
30. a[k--] = a[j];
31. }
32. k = count;
33. **for**(**int** j = 0 ; j < m ; j ++){
34. a[j] = a[k++];
35. }
36. **for**(**int** j = 0 ; j < count ; j ++){
37. System.out.print(a[j]+" ");
38. }
39. }
40. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. **import** java.util.Scanner;
5. /\*\*
6. \* Created by sakura on 16/6/5.
7. \*/
8. /\*\*
9. \*      编写一个函数，输入n为偶数时，调用函数求1/2+1/4+...+1/n,当输入n为奇数时，调用函数   1/1+1/3+...+1/n(利用指针函数)
10. \*
11. \* \*/
12. **public** **class** Test6 {
13. **public** **static** **void** main(String[] args){
14. Scanner in = **new** Scanner(System.in);
15. **double** n = in.nextInt();
16. **double** total = 0;
17. **if**(n % 2 == 0){
18. **while**(n > 0){
19. total += 1/n;
20. System.out.print("1"+"/"+n+" ");
21. n -= 2;
22. }
23. System.out.println(total);
24. }**else** **if**(n % 2 != 0){
25. **while**(n > 0){
26. total += 1/n;
27. System.out.print("1"+"/"+n+" ");
28. n -= 3;
29. }
30. System.out.println(total + 1);
31. }**else**{
32. System.out.print(n);
33. }
34. }
35. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. **import** java.util.Scanner;
5. /\*\*
6. \* Created by sakura on 16/6/1.
7. \*/
8. /\*\*
9. \*
10. \*  有n个人围成一圈，顺序排号。从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的那位。
11. \* \*/
12. **public** **class** Test7 {
13. **public** **static** **void** main(String[] args){
14. Scanner in = **new** Scanner(System.in);
15. **int** n = in.nextInt();
16. **int** a[] = **new** **int**[100];
17. **for**(**int** i = 0;i < n ;i++){
18. a[i] = i + 1;
19. }
20. **int** i = 0;
21. **int** count = 1;
22. **int** newN = n;
23. **int** j = 0;
24. **int** time = 0;
25. **while**(newN >= 3){
26. i = 0;
27. **while**(a[i] != '\0'){
28. **if**(count ==3){
29. j = i;
30. **while**(a[j] != '\0'){
31. a[j] = a[j + 1];
32. j++;
33. }
34. newN--;
35. }**else**
36. i++;
37. **if**(count == 3)
38. count = 1;
39. **else**
40. count++;
41. }
42. }
43. **int** k = 0 ;
44. **while**(a[k] != '\0'){
45. System.out.print(a[k++]+" ");
46. }
47. }
48. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/6/7.
5. \*/
6. /\*\*
7. \*  题目：809\*??=800\*??+9\*??+1
8. \*   其中??代表的两位数,8\*??的结果为两位数，9\*??的结果为3位数。求??代表的两位数，及809\*??后的结果。
9. \*
10. \* \*/
11. **public** **class** Test8 {
12. **public** **static** **void** main(String[] args){
13. **int** i;
14. **for**(i = 10 ; i <= 99 ; i ++){
15. **if**(8 \* i >= 10 && 8 \* i <= 99)
16. **if**(9 \* i >= 100 && 9 \* i <= 999)
17. **if**(809 \* i == 800 \* i + 9 \* i + 1)  //这个题目无解  除非去掉算式最后的+1
18. **break**;
19. }
20. System.out.println(i);
21. }
22. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/6/14.
5. \*/
6. /\*\*
7. \*  求0—7所能组成的奇数个数。
8. \*
9. \*      以下代码为 0-7所能组成的7位奇数
10. \*      未能完成题目,还需求6位奇数
11. \* \*/
12. **public** **class** Test9 {
13. **public** **static** **final** **int** lim = 7;
14. **public** **static** **void** main(String[] args){
15. **int** i = 0;
16. **int** count = 0;
17. **while**(i <= lim){
18. **if**(i == 0){
20. }**else** **if**(i % 2 != 0){
21. count += Fun(lim + 1 - 1) - Fun(lim + 1 - 2);    //题目为0 - 7 所以要 lim + 1
22. }
23. i ++;
24. }
25. System.out.println(count);
26. }
27. **public** **static** **int** Fun(**int** a){
28. **int** i = 1;
29. **int** t = 1;
30. **if**(a == 1)
31. **return** 1;
32. **while**(i <= a){
33. t = i \* t;
34. i++;
35. }
36. **return** t;
37. }
38. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/28.
5. \*/
6. /\*\*
7. \*  求1+2!+3!+...+20!的和
8. \*
9. \*  注:0! = 1
10. \*
11. \*  未测试结果
12. \*  测试结果为268040729
13. \* \*/
14. **public** **class** TestJieCheng {
15. **public** **static** **void** main(String[] atgs){
16. **int** total = 0;
17. **for**(**int** i = 1;i <= 20;i++){
18. total += JieCheng(i);
19. }
20. System.out.print(total);
21. }
22. **public** **static** **int** JieCheng(**int** a){
23. **int** i = 1;
24. **int** total = 1;
25. **while**(i <= a){
26. total = total \* i;
27. i++;
28. }
29. **return** total;
30. }
31. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/28.
5. \*/
6. /\*\*
7. \*
8. \*  求一个3\*3矩阵对角线元素之和
9. \*
10. \* \*/
11. **public** **class** ThreeArray {
12. **public** **static** **void** main(String[] args) {
13. **int**[][] threeArray = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
14. **int** total1 = 0;
15. **int** total2 = 0;
16. **int** i, j;
17. **for**( i = 0,j = 0;i < 3 && j < 3;i++,j++) {
18. total1 += threeArray[i][j];
19. }
20. **for**(i = 0 ,j = 2;i < 3 && j >= 0;i++,j--){
21. total2 += threeArray[i][j];
22. }
23. System.out.println(total1+" "+total2);
24. }
26. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/11.
5. \*/
6. /\*\*
7. \*
8. \* 题目：输入两个正整数m和n，求其最大公约数和最小公倍数
9. \*
10. \*
11. \*               答案正确
12. \*              求最大公约数使用google的辗除法,未理解,记住即可,涉及欧几里\*\*,
13. \*
14. \* \*/
15. **public** **class** TwoNumberFunction {
16. **public** **static** **void** main(String[] args){
17. System.out.println(calculateMax(7,102));
18. System.out.println(calculateMin(10,101));
19. }
20. **public** **static** **int** calculateMax(**int** a,**int** b){
21. **int** time = -1;
22. **if**(a>b){
23. **while**(time != 0){               //利用辗除法(记住即可)
24. time = a%b;
25. a = b;
26. b = time;
27. }
28. **return** a;
29. }**else**{
30. **while**(time != 0){
31. time = b%a;
32. b = a;
33. a = time;
34. }
35. **return** b;
36. }
37. }
38. **public** **static** **int** calculateMin(**int** a,**int** b){
39. **int** time;
40. **if**(a>b){
41. time = a;
42. **while**(time%a != 0 || time%b !=0){
43. time++;
44. }
45. **return** time;
46. }**else**{
47. time = b;
48. **while**(time%a != 0 || time%b != 0 ){
49. time++;
50. }
51. **return** time;
52. }
53. }
54. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/14.
5. \*/
7. /\*\*
8. \* 题目：一个数如果恰好等于它的因子之和，这个数就称为"完数"。例如6=1＋2＋3.编程 找出1000以内的所有完数
9. \*
10. \* \*/
11. **public** **class** WanShu {
12. **public** **static** **void** main(String [] args){
13. **int** total = 0;
14. **int** time = 1;
15. **int** time1 = 0;
16. **int** forTime;
17. **while**(time <= 1000){
18. **for**(forTime = 1;forTime < time;forTime++){      //不能forTime <= time ,不能加上 = ;
19. **if**(time%forTime == 0){
20. total=total+forTime;
21. }
22. }
24. **if**(time == total){
25. System.out.println(time+" 是完数!");
26. }
27. time++;
28. total = 0;
29. }
30. }
31. }

**[java]** [view plain](http://blog.csdn.net/sakura_yuan/article/details/51676059) [copy](http://blog.csdn.net/sakura_yuan/article/details/51676059)

1. **package** package01;
3. /\*\*
4. \* Created by sakura on 16/5/30.
5. \*/
6. /\*\*
7. \*
8. \*     1            5
9. \*    1 1           4
10. \*   1 2 1          3
11. \*  1 3 3 1         2
12. \* 1 4 6 4 1       1
13. \*1 5 10 10 5 1     0
14. \* \*/
15. **public** **class** YnagHui {
16. **public** **static** **void** main(String[] args){
17. **int** row = 10;
18. **int** timeRow = 1;
19. **int** spaceCount = row - 1;
20. **int** numberCount = 1;
21. **int**[] number = **new** **int**[100];
22. **int**[] newNumber = **new** **int**[100];
23. newNumber[0] = 1;
24. number[0] = 1; number[1] = 1;
25. **int** m,n;
26. **while**(timeRow <= row){
27. **for**(**int** i = 0;i < spaceCount;i++)
28. System.out.print(" ");
29. spaceCount --;
30. **if**(numberCount == 1)
31. System.out.print("1 ");
32. **else** **if**(numberCount == 2)
33. System.out.print("1 1 ");
34. **else**{
35. System.out.print("1 ");
36. m = 0;n = timeRow - 2;
37. **while**((m + 1) <= n){
38. System.out.print(number[m] + number[m + 1]+" ");
39. newNumber[m + 1] =  number[m] + number[m + 1];      // 1
40. m++;
42. }
43. newNumber[n + 1] = 1;
44. System.out.print("1 ");
45. number = newNumber.clone();                     //难点:不能使用number = newNumber,这是按引用传递,在1处会出错
46. }                                                   //(在使用1后不是单单对newNumber进行了操作,对number数组也进行了操作,)
47. //.clone为复制数组,number的操作不会影响newNumber数组
48. numberCount ++;
49. timeRow ++;
50. System.out.print("\n");
51. }
52. }
53. }