



第二版：Java 算法 50 道

目录

第二版：Java 算法 50 道	1
1.指数计算问题	2
2.指定范围包含的素数	3
3.水仙花数	4
4.分解质因数	5
5.条件运算符使用	6
6.公约数和公倍数	7
7.统计字符串中类型个数	8
8.求 $s = a + aa + aaa + aaaa + aa...a$ 的值	11
9.指定范围的完数	12
10.反指数计算	13
11.组合	14
12.梯度计算	15
13.求未知数	17
14.日期计算	18
15.排序	19
16.冒泡排序	20
17.反推计算	21
18.数组计算	22
19.打印出如下图案（菱形）	23
20.数列求和	24
21.求 $1 + 2! + 3! + \dots + 20!$ 的和	25
22.利用递归方法求 $5!$	25
23.递归计算	26
24.倒序打印	27
25.回文数	27
26.匹配单词	29
27.求 100 之内的素数	32



28.对 10 个数进行排序	32
29.求一个 3*3 矩阵对角线元素之和	33
30.比较排序	34
31.将一个数组逆序输出。	35
32.取一个整数 a 从右端开始的 4~7 位。	36
33.打印出杨辉三角形 (要求打印出 10 行如下图)	37
34.输入 3 个数 a,b,c, 按大小顺序输出。	38
35.选择排序	39
36.交换位置	40
37.排序问题	41
38.计算字符串总长度	42
39.求和	43
40.字符串排序。	44
41.递归	45
42.809??=800??+9*??+1	45
43.求 0—7 所能组成的奇数个数。	46
44.一个偶数总能表示为两个素数之和。	47
45.判断一个素数能被几个 9 整除	48
46.两个字符串连接程序	49
47.打印练习	49
48.加密算法	50
49.计算字符串中子串出现的次数	51
50.求平均数	51

我们的网站: <https://tech.souyunku.com>

关注我们的公众号：搜云库技术团队，回复以下关键字

回复:【进群】邀请您进「技术架构分享群」

回复:【内推】即可进: 北京, 上海, 广州, 深圳, 杭州, 成都, 武汉, 南京,

✧ 微信搜一搜

🔍 搜云库技术团队



郑州, 西安, 长沙「程序员工作内推群」

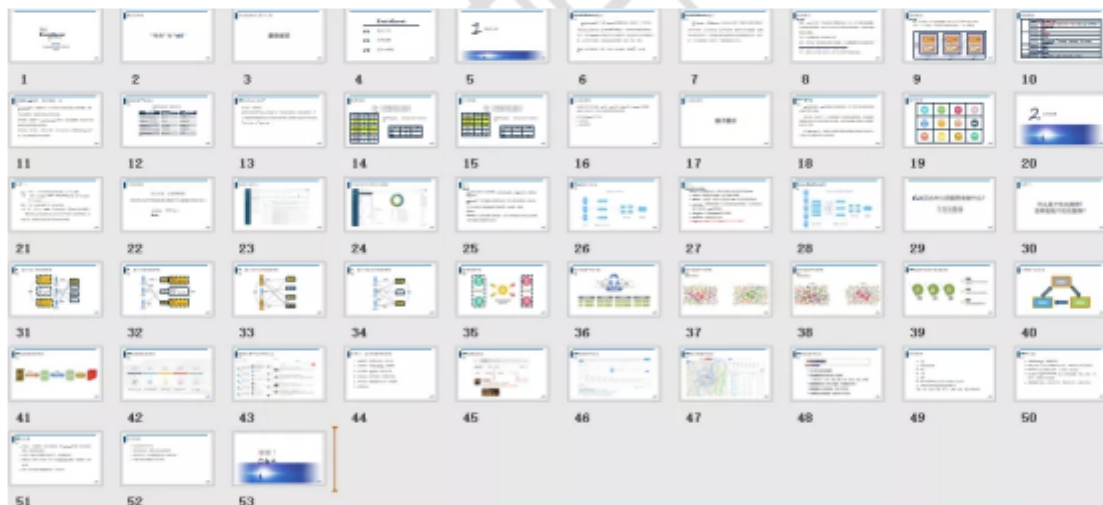
回复 **【1024】** 送 4000G 最新架构师视频

回复 **【PPT】** 即可无套路获取, 以下最新整理调优 PPT!

46 页《JVM 深度调优, 演讲 PPT》



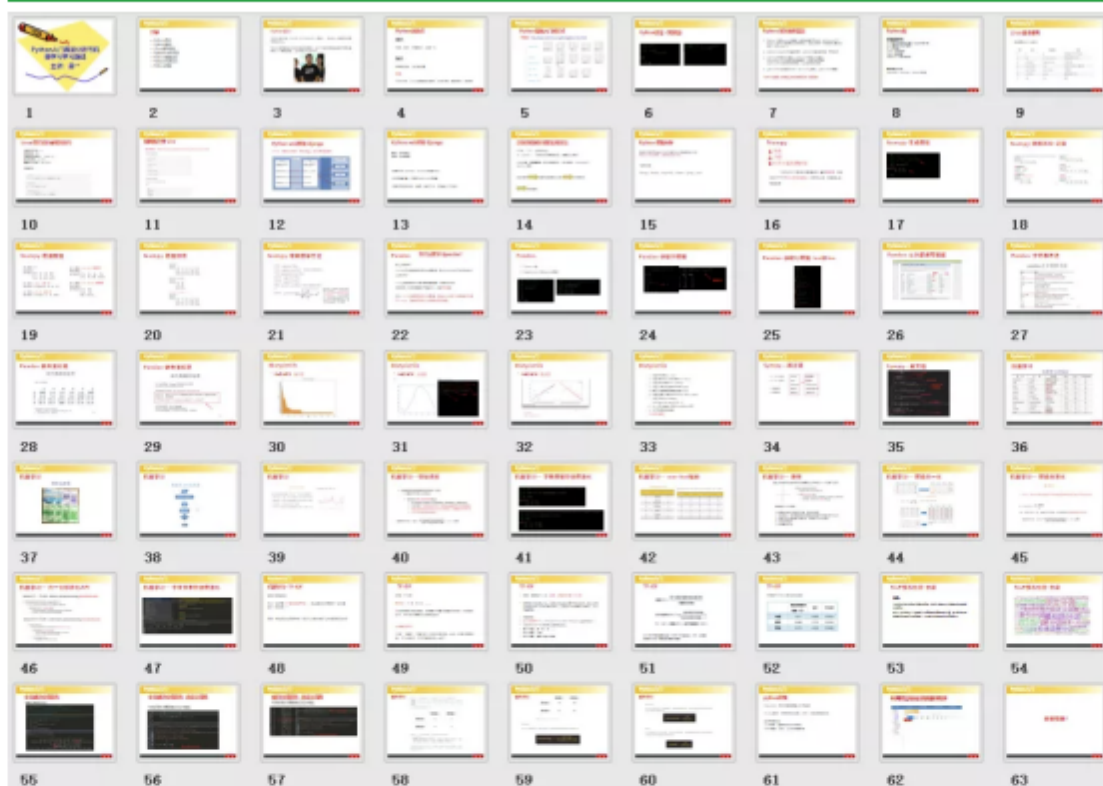
53 页《Elasticsearch 调优演讲 PPT》



63 页《Python 数据分析入门 PPT》

微信搜一搜

搜云库技术团队



微信扫一扫

<https://tech.souyunku.com>

技术、架构、资料、工作、内推
专注于分享最有价值的互联网技术干货文章

1.指数计算问题



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，问每个月的兔子对数为多少？

程序分析：兔子的规律为数列 1,1,2,3,5,8,13,21....

```
public class Prog1{
    public static void main(String[] args){
        int n = 10;
        System.out.println("第"+n+"个月兔子总数为"+fun(n));
    }
    private static int fun(int n){
        if(n==1 || n==2)
            return 1;
        else
            return fun(n-1)+fun(n-2);
    }
}
```

2.指定范围包含的素数

判断 101-200 之间有多少个素数，并输出所有素数。

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。

```
public class Prog2{
    public static void main(String[] args){
        int m = 1;
        int n = 1000;
        int count = 0;
```



```
//统计素数个数
for(int i=m;i<n;i++){
    if(isPrime(i)){
        count++;
        System.out.print(i+" ");
        if(count%10==0){
            System.out.println();
        }
    }
}
System.out.println();
System.out.println("在"+m+"和"+n+"之间共有"+count+"个素数");
}

//判断素数
private static boolean isPrime(int n){
    boolean flag = true;
    if(n==1)
        flag = false;
    else{
        for(int i=2;i<=Math.sqrt(n);i++){
            if((n%i)==0 || n==1){
                flag = false;
                break;
            }
        }
        else
            flag = true;
    }
}
return flag;
}
```



}

3.水仙花数

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 $153 = 1^3 + 5^3 + 3^3$ 。

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。

```
public class Prog3{
    public static void main(String[] args){
        for(int i=100;i<1000;i++){
            if(isLotus(i))
                System.out.print(i+" ");
        }
        System.out.println();
    }
    //判断水仙花数
    private static boolean isLotus(int lotus){
        int m = 0;
        int n = lotus;
        int sum = 0;
        m = n/100;
        n -= m*100;
        sum = m*m*m;
        m = n/10;
        n -= m*10;
        sum += m*m*m + n*n*n;
        if(sum==lotus)
```




```
        return true;
    else
        return false;
    }
}
```

4.分解质因数

将一个正整数分解质因数。例如：输入 90,打印出 $90=2 \times 3 \times 3 \times 5$ 。

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 $n \neq k$ ，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 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。

```
public class Prog4{
    public static void main(String[] args){
        int n = 13;
        decompose(n);
    }
    private static void decompose(int n){
        System.out.print(n+"=");
        for(int i=2;i<n+1;i++){
            while(n%i==0 && n!=i){
                n/=i;
                System.out.print(i+"*");
            }
        }
    }
}
```




```

        if(n==i){
            System.out.println(i);
            break;
        }
    }
}
}
}

```

5.条件运算符使用

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。

程序分析：(a>b)?a:b 这是条件运算符的基本例子。

```

public class Prog5{
    public static void main(String[] args){
        int n = -1;
        try{
            n = Integer.parseInt(args[0]);
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("请输入成绩");
            return;
        }
        grade(n);
    }
    //成绩等级计算
    private static void grade(int n){
        if(n>100 || n<0)
            System.out.println("输入无效");
    }
}

```



```

        else{
            String str = (n>=90)?"分, 属于 A 等":((n>60)?"分, 属于 B 等":
分, 属于 C 等");
            System.out.println(n+str);
        }
    }
}

```

6.公约数和公倍数

输入两个正整数 m 和 n , 求其最大公约数和最小公倍数。

程序分析: 利用辗除法。

```

public class Prog6{
    public static void main(String[] args){
        int m,n;
        try{
            m = Integer.parseInt(args[0]);
            n = Integer.parseInt(args[1]);
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("输入有误");
            return;
        }
        max_min(m,n);
    }
    //求最大公约数和最小公倍数
    private static void max_min(int m, int n){
        int temp = 1;
        int yshu = 1;
    }
}

```



```

int bshu = m*n;
if(n<m){
    temp = n;
    n = m;
    m = temp;
}
while(m!=0){
    temp = n%m;
    n = m;
    m = temp;
}
yshu = n;
bshu /= n;
System.out.println(m+"和"+n+"的最大公约数为"+yshu);
System.out.println(m+"和"+n+"的最小公倍数为"+bshu);
}
}

```

7.统计字符串中类型个数

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。

程序分析：利用 while 语句,条件为输入的字符不为 '\n'.

```

import java.util.Scanner;
public class Prog7_1{
    public static void main(String[] args){
        System.out.print("请输入一串字符: ");
        Scanner scan = new Scanner(System.in);
        String str = scan.nextLine();//将一行字符转化为字符串
    }
}

```



```

        scan.close();
        count(str);
    }
    //统计输入的字符数
    private static void count(String str){
        String E1 = "[\u4e00-\u9fa5]";//汉字
        String E2 = "[a-zA-Z]";
        String E3 = "[0-9]";
        String E4 = "\\s";//空格
        int countChinese = 0;
        int countLetter = 0;
        int countNumber = 0;
        int countSpace = 0;
        int countOther = 0;
        char[] array_Char = str.toCharArray();//将字符串转化为字符数组
        String[] array_String = new String[array_Char.length];//汉字只能
        作为字符串处理
        for(int i=0;i<array_Char.length;i++)
            array_String[i] = String.valueOf(array_Char[i]);
        //遍历字符串数组中的元素
        for(String s:array_String){
            if(s.matches(E1))
                countChinese++;
            else if(s.matches(E2))
                countLetter++;
            else if(s.matches(E3))
                countNumber++;
            else if(s.matches(E4))
                countSpace++;
            else
                countOther++;
        }
    }
}

```



```

    }
    System.out.println("输入的汉字个数: "+countChinese);
    System.out.println("输入的字母个数: "+countLetter);
    System.out.println("输入的数字个数: "+countNumber);
    System.out.println("输入的空格个数: "+countSpace);
    System.out.println("输入的其它字符个数: "+countSpace);
}
}

import java.util.*;
public class Prog7_2{
    public static void main(String[] args){
        System.out.println("请输入一行字符: ");
        Scanner scan = new Scanner(System.in);
        String str = scan.nextLine();
        scan.close();
        count(str);
    }
    //统计输入的字符
    private static void count(String str){
        List<String> list = new ArrayList<String>();
        char[] array_Char = str.toCharArray();
        for(char c:array_Char)
            list.add(String.valueOf(c)); //将字符作为字符串添加到 list 表中
        Collections.sort(list); //排序
        for(String s:list){
            int begin = list.indexOf(s);
            int end = list.lastIndexOf(s);
            //索引结束统计字符数
            if(list.get(end) == s)

```



```

        System.out.println("字符 '"+s+"' 有"+(end-begin+1)+"个
");
    }
}
}
}

```

8.求 $s=a+aa+aaa+aaaa+aa...a$ 的值

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。

程序分析：关键是计算出每一项的值。

```

import java.util.Scanner;

public class Prog8{
    public static void main(String[] args){
        System.out.print("求 s=a+aa+aaa+aaaa+...的值，请输入 a 的值：
");
        Scanner scan = new Scanner(System.in).useDelimiter("\\s*");//以
        空格作为分隔符
        int a = scan.nextInt();
        int n = scan.nextInt();
        scan.close();//关闭扫描器
        System.out.println(expressed(2,5)+add(2,5));
    }
    //求和表达式
    private static String expressed(int a,int n){
        StringBuffer sb = new StringBuffer();
        StringBuffer subSB = new StringBuffer();
    }
}

```



```

        for(int i=1;i<n+1;i++){
            subSB = subSB.append(a);
            sb = sb.append(subSB);
            if(i<n)
                sb = sb.append("+");
        }
        sb.append("=");
        return sb.toString();
    }
    //求和
    private static long add(int a,int n){
        long sum = 0;
        long subSUM = 0;
        for(int i=1;i<n+1;i++){
            subSUM = subSUM*10+a;
            sum = sum+subSUM;
        }
        return sum;
    }
}

```

9.指定范围的完数

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。

```

public class Prog9{
    public static void main(String[] args){
        int n = 10000;
        compNumber(n);
    }
}

```




```

    }
    //求完数
    private static void compNumber(int n){
        int count = 0;
        System.out.println(n+"以内的完数: ");
        for (int i = 1; i <= 10000; i++) {
            int temp = 0; // 定义因子之和变量
            for (int n = 1; n < i / 2 + 1; n++) {
                if (i % n == 0) {
                    temp += n; // 能被整除的除数则被加到 temp 中
                }
            }
            if (temp == i) { // 如果因子之和与原数相等的话，说明是完数
                System.out.println(i + " "); // 输出完数
            }
        }
    }
}

```

10.反指数计算

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？

```

import java.util.Scanner;
public class Prog10{
    public static void main(String[] args){
        System.out.print("请输入小球落地时的高度和求解的次数: ");
        Scanner scan = new Scanner(System.in).useDelimiter("\\s");
        int h = scan.nextInt();
    }
}

```



```

        int n = scan.nextInt();
        scan.close();
        distance(h,n);
    }

    //小球从 h 高度落下, 经 n 次反弹后经过的距离和反弹的高度
    private static void distance(double h,int n){
        double length = 0;
        for(int i=0;i<n;i++){
            length += h;
            h *=0.5 ;
            length += h;
        }
        System.out.println("经过第"+n+"次反弹后, 小球共经过"+length+"
        米, "+"第"+n+"次反弹高度为"+h+"米");
    }
}

```

11.组合

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?

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。

```

public class Prog11{
    public static void main(String[] args){
        int count = 0;
        int n = 0;
        for(int i=1;i<5;i++){

```



```

        for(int j=1;j<5;j++){
            if(j==i)
                continue;
            for(int k=1;k<5;k++){
                if(k!=i && k!=j){
                    n = i*100+j*10+k;
                    System.out.print(n+" ");
                    if(++count)%5==0)
                        System.out.println();
                }
            }
        }
        System.out.println();
        System.out.println("符合条件的数共: "+count+"个");
    }
}

```

12.梯度计算

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，可按 7.5%提成；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？

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。

```
import java.io.*;
```



```
public class Prog12{
    public static void main(String[] args){
        System.out.print("请输入当前利润: ");
        long profit = Long.parseLong(key_Input());
        System.out.println("应发奖金: "+bonus(profit));
    }
    //接受从键盘输入的内容
    private static String key_Input(){
        String str = null;
        BufferedReader bufIn = new BufferedReader(new
InputStreamReader(System.in));
        try{
            str = bufIn.readLine();
        }catch(IOException e){
            e.printStackTrace();
        }finally{
            try{
                bufIn.close();
            }catch(IOException e){
                e.printStackTrace();
            }
        }
        return str;
    }
    //计算奖金
    private static long bonus(long profit){
        long prize = 0;
        long profit_sub = profit;
        if(profit>1000000){
            profit = profit_sub-1000000;
            profit_sub = 1000000;
        }
    }
}
```



```

        prize += profit*0.01;
    }
    if(profit>600000){
        profit = profit_sub-600000;
        profit_sub = 600000;
        prize += profit*0.015;
    }
    if(profit>400000){
        profit = profit_sub-400000;
        profit_sub = 400000;
        prize += profit*0.03;
    }
    if(profit>200000){
        profit = profit_sub-200000;
        profit_sub = 200000;
        prize += prize*0.05;
    }
    if(profit>100000){
        profit = profit_sub-100000;
        profit_sub = 100000;
        prize += profit*0.075;
    }
    prize += profit_sub*0.1;
    return prize;
}
}

```

13.求未知数



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？

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。

```
public class Prog13{
    public static void main(String[] args){
        int n=0;
        for(int i=0;i<100001;i++){
            if(isCompSqrt(i+100) && isCompSqrt(i+268)){
                n = i;
                break;
            }
        }
        System.out.println("所求的数是: "+n);
    }
    //判断完全平方数
    private static boolean isCompSqrt(int n){
        boolean isComp = false;
        for(int i=1;i<Math.sqrt(n)+1;i++){
            if(n==Math.pow(i,2)){
                isComp = true;
                break;
            }
        }
        return isComp;
    }
}
```



14.日期计算

输入某年某月某日，判断这一天是这一年的第几天？

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。

```
import java.util.Scanner;
public class Prog14{
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in).useDelimiter("\\D");//匹配非数字
        System.out.print("请输入当前日期（年-月-日）：");
        int year = scan.nextInt();
        int month = scan.nextInt();
        int date = scan.nextInt();
        scan.close();
        System.out.println("今天是"+year+"年的第"+analysis(year,month,date)+"天");
    }
    //判断天数
    private static int analysis(int year, int month, int date){
        int n = 0;
        int[] month_date = new int[]
{0,31,28,31,30,31,30,31,31,30,31,30};
        if((year%400)==0 || ((year%4)==0)&&((year%100)!=0))
            month_date[2] = 29;
        for(int i=0;i<month;i++)
            n += month_date[i];
        return n+date;
    }
}
```




```
}  
}
```

15.排序

输入三个整数 x, y, z ，请把这三个数由小到大输出。

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。

```
import java.util.Scanner;  
public class Prog15{  
    public static void main(String[] args){  
        Scanner scan = new Scanner(System.in).useDelimiter("\\D");  
        System.out.print("请输入三个数：");  
        int x = scan.nextInt();  
        int y = scan.nextInt();  
        int z = scan.nextInt();  
        scan.close();  
        System.out.println("排序结果：" + sort(x,y,z));  
    }  
    //比较两个数的大小  
    private static String sort(int x,int y,int z){  
        String s = null;  
        if(x>y){  
            int t = x;  
            x = y;  
            y = t;  
        }  
    }  
}
```



```

        if(x>z){
            int t = x;
            x = z;
            z = t;
        }
        if(y>z){
            int t = z;
            z = y;
            y = t;
        }
        s = x+" "+y+" "+z;
        return s;
    }
}

```

16.冒泡排序

输出 9*9 口诀。

程序分析：分行与列考虑，共 9 行 9 列，i 控制行，j 控制列。

```

public class Prog16{
    public static void main(String[] args){
        for(int i=1;i<10;i++){
            for(int j=1;j<i+1;j++){
                System.out.print(j+"*"+i+"="+j*i+" ");
            }
            System.out.println();
        }
    }
}

```



17.反推计算

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。

程序分析：采取逆向思维的方法，从后往前推断。

```
public class Prog17{
    public static void main(String[] args){
        int m = 1;
        for(int i=10;i>0;i--){
            m = 2*m + 2;
            System.out.println("小猴子共摘了"+m+"桃子");
        }
    }
}
```

18.数组计算

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，请编程找出三队赛手的名单。

```
import java.util.ArrayList;
public class Prog18{
    String a,b,c;//甲队成员
    public static void main(String[] args){
```



```
String[] racer = {"x","y","z"};//乙队成员
ArrayList<Prog18> arrayList = new ArrayList<Prog18>();
for(int i=0;i<3;i++)
    for(int j=0;j<3;j++)
        for(int k=0;k<3;k++){
            Prog18 prog18 = new Prog18(racer[i],racer[j],racer[k]);
            if(!prog18.a.equals(prog18.b)
&& !prog18.a.equals(prog18.c) && !prog18.b.equals(prog18.c) &&
                !prog18.a.equals("x") && !prog18.c.equals("x")
&& !prog18.c.equals("z"))
                arrayList.add(prog18);
        }
    for(Object obj:arrayList)
        System.out.println(obj);
}
//构造方法
private Prog18(String a,String b,String c){
    this.a = a;
    this.b = b ;
    this.c = c;
}
public String toString(){
    return "a的对手是"+a+"  "+"b的对手是"+b+"  "+"c的对手是
"+c;
}
}
```

19.打印出如下图案（菱形）

*



```

    ***
  *****
*****
  *****
    ***
      *
  
```

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。

```

public class Prog19{
    public static void main(String[] args){
        int n = 5;
        printStar(n);
    }
    //打印星星
    private static void printStar(int n){
        //打印上半部分
        for(int i=0;i<n;i++){
            for(int j=0;j<2*n;j++){
                if(j<n-i)
                    System.out.print(" ");
                if(j>=n-i && j<=n+i)
                    System.out.print("*");
            }
            System.out.println();
        }
        //打印下半部分
        for(int i=1;i<n;i++){
            System.out.print(" ");
            for(int j=0;j<2*n-i;j++){
  
```



```

        if(j<i)
            System.out.print(" ");
        if(j>=i && j<2*n-i-1)
            System.out.print("*");
        }
        System.out.println();
    }
}
}

```

20.数列求和

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。

程序分析：请抓住分子与分母的变化规律。

```

public class Prog20{
    public static void main(String[] args){
        double n1 = 1;
        double n2 = 1;
        double fraction = n1/n2;
        double Sn = 0;
        for(int i=0;i<20;i++){
            double t1 = n1;
            double t2 = n2;
            n1 = t1+t2;
            n2 = t1;
            fraction = n1/n2;
            Sn += fraction;
        }
    }
}

```



```
    }
    System.out.print(Sn);
}
}
```

21.求 $1+2!+3!+\dots+20!$ 的和

程序分析：此程序只是把累加变成了累乘。

```
public class Prog21{
    public static void main(String[] args){
        long sum = 0;
        for(int i=0;i<20;i++){
            sum += factorial(i+1);
            System.out.println(sum);
        }
        //阶乘
        private static long factorial(int n){
            int mult = 1;
            for(int i=1;i<=n;i++){
                mult *= i;
            }
            return mult;
        }
    }
}
```

22.利用递归方法求 $5!$ 。

程序分析：递归公式： $fn = fn_{-1} * 4!$



```
public class Prog22{
    public static void main(String[] args){
        System.out.println(fact(10));
    }
    //递归求阶乘
    private static long fact(int n){
        if(n==1)
            return 1;
        else
            return fact(n-1)*n;
    }
}
```

23.递归计算

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， he 说是 10 岁。请问第五个人多大？

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。

```
public class Prog23{
    public static void main(String[] args){
        System.out.println(getAge(5,2));
    }
    //求第 m 位同志的年龄
    private static int getAge(int m,int n){
        if(m==1)
            return 10;
    }
}
```



```

        else
            return getAge(m-1,n)+n;
    }
}

```

24.倒序打印

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。

```

public class Prog24{
    public static void main(String[] args){
        int n = Integer.parseInt(args[0]);
        int i = 0;
        int[] a = new int[5];
        do{
            a[i] = n%10;
            n /= 10;
            ++i;
        }while(n!=0);
        System.out.print("这是一个"+i+"位数，从个位起，各位数字依次为：");
        for(int j=0;j<i;j++)
            System.out.print(a[j]+" ");
    }
}

```

25.回文数



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。

```
import java.io.*;
public class Prog25{
    public static void main(String[] args){
        int n = 0;
        System.out.print("请输入一个 5 位数: ");
        BufferedReader bufin = new BufferedReader(new
InputStreamReader(System.in));
        try{
            n = Integer.parseInt(bufin.readLine());
        }catch(IOException e){
            e.printStackTrace();
        }finally{
            try{
                bufin.close();
            }catch(IOException e){
                e.printStackTrace();
            }
        }
        palin(n);
    }
    private static void palin(int n){
        int m = n;
        int[] a = new int[5];
        if(n<10000 || n>99999){
            System.out.println("输入的不是 5 位数! ");
            return;
        }else{
            for(int i=0;i<5;i++){
```



```

        a[i] = n%10;
        n /= 10;
    }
    if(a[0]==a[4] && a[1]==a[3])
        System.out.println(m+"是一个回文数");
    else
        System.out.println(m+"不是回文数");
    }
}
}

```

26.匹配单词

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。

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。

```

import java.io.*;
public class Prog26{
    public static void main(String[] args){
        String str = new String();
        BufferedReader bufIn = new BufferedReader(new
        InputStreamReader(System.in));
        System.out.print("请输入星期的英文单词前两至四个字母) : ");
        try{
            str = bufIn.readLine();
        }catch(IOException e){
            e.printStackTrace();
        }
    }
}

```



```

    }finally{
        try{
            bufIn.close();
        }catch(IOException e){
            e.printStackTrace();
        }
    }
    week(str);
}

private static void week(String str){
    int n = -1;
    if(str.trim().equalsIgnoreCase("Mo") ||
str.trim().equalsIgnoreCase("Mon") ||
str.trim().equalsIgnoreCase("Mond"))
        n = 1;
    if(str.trim().equalsIgnoreCase("Tu") ||
str.trim().equalsIgnoreCase("Tue") || str.trim().equalsIgnoreCase("Tues"))
        n = 2;
    if(str.trim().equalsIgnoreCase("We") ||
str.trim().equalsIgnoreCase("Wed") ||
str.trim().equalsIgnoreCase("Wedn"))
        n = 3;
    if(str.trim().equalsIgnoreCase("Th") ||
str.trim().equalsIgnoreCase("Thu") || str.trim().equalsIgnoreCase("Thur"))
        n = 4;
    if(str.trim().equalsIgnoreCase("Fr") ||
str.trim().equalsIgnoreCase("Fri") || str.trim().equalsIgnoreCase("Frid"))
        n = 5;
    if(str.trim().equalsIgnoreCase("Sa") ||
str.trim().equalsIgnoreCase("Sat") || str.trim().equalsIgnoreCase("Satu"))
        n = 2;
}

```



```

        if(str.trim().equalsIgnoreCase("Su") ||
str.trim().equalsIgnoreCase("Sun") || str.trim().equalsIgnoreCase("Sund"))
        n = 0;
        switch(n){
            case 1:
                System.out.println("星期一");
                break;
            case 2:
                System.out.println("星期二");
                break;
            case 3:
                System.out.println("星期三");
                break;
            case 4:
                System.out.println("星期四");
                break;
            case 5:
                System.out.println("星期五");
                break;
            case 6:
                System.out.println("星期六");
                break;
            case 0:
                System.out.println("星期日");
                break;
            default:
                System.out.println("输入有误! ");
                break;
        }
    }
}

```



27.求 100 以内的素数

```
public class Prog27{
    public static void main(String[] args){
        int n = 100;
        System.out.print(n+"以内的素数: ");
        for(int i=2;i<n+1;i++){
            if(isPrime(i))
                System.out.print(i+" ");
        }
    }
    //求素数
    private static boolean isPrime(int n){
        boolean flag = true;
        for(int i=2;i<Math.sqrt(n)+1;i++){
            if(n%i==0){
                flag = false;
                break;
            }
        }
        return flag;
    }
}
```

28.对 10 个数进行排序

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。



```
public class Prog28{
    public static void main(String[] args){
        int[] a = new int[]{31,42,21,50,12,60,81,74,101,93};
        for(int i=0;i<10;i++){
            for(int j=0;j<a.length-i-1;j++){
                if(a[j]>a[j+1]){
                    int temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                }
            }
        }
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
    }
}
```

29.求一个 3*3 矩阵对角线元素之和

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。

```
public class Prog29{
    public static void main(String[] args){
        int[][] a = new int[][] {{100,2,3},{4,5,6},{17,8,9}};
        matrSum(a);
    }
    private static void matrSum(int[][] a){
        int sum1 = 0;
        int sum2 = 0;
```



```

        for(int i=0;i<a.length;i++)
            for(int j=0;j<a[i].length;j++){
                if(i==j) sum1 += a[i][j];
                if(j==a.length-i-1) sum2 += a[i][j];
            }
        System.out.println("矩阵对角线之和分别是："+sum1+"和"+sum2);
    }
}

```

30.比较排序

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。

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。

```

import java.util.Scanner;
public class Prog30{
    public static void main(String[] args){
        int[] A = new int[]{0,8,7,5,9,1,2,4,3,12};
        int[] B = sort(A);
        print(B);
        System.out.println();
        System.out.print("请输入 10 个数的数组: ");
        Scanner scan = new Scanner(System.in);
        int a = scan.nextInt();
        scan.close();
        int[] C = insert(a,B);
        print(C);
    }
}

```



//选择排序

```
private static int[] sort(int[] A){
    int[] B = new int[A.length];
    for(int i=0;i<A.length-1;i++){
        int min = A[i];
        for(int j=i+1;j<A.length;j++){
            if(min>A[j]){
                int temp = min;
                min = A[j];
                A[j] = temp;
            }
            B[i] = min;
        }
    }
    B[A.length-1] = A[A.length-1];
    return B;
}
```

//打印

```
private static void print(int[] A){
    for(int i=0;i<A.length;i++)
        System.out.print(A[i]+" ");
}
```

//插入数字

```
private static int[] insert(int a,int[] A){
    int[] B = new int[A.length+1];
    for(int i=A.length-1;i>0;i--){
        if(a>A[i]){
            B[i+1] = a;
            for(int j=0;j<=i;j++)
                B[j] = A[j];
            for(int k=i+2;k<B.length;k++)
```



```

        B[k] = A[k-1];
        break;
    }
    return B;
}
}

```

31.将一个数组逆序输出。

程序分析：用第一个与最后一个交换。

```

public class Prog31{
    public static void main(String[] args){
        int[] A = new int[]{1,2,3,4,5,6,7,8,9};
        print(A);
        System.out.println();
        int[] B = reverse(A);
        print(B);
    }
    private static int[] reverse(int[] A){
        for(int i=0;i<A.length/2;i++){
            int temp = A[A.length-i-1];
            A[A.length-i-1] = A[i];
            A[i] = temp;
        }
        return A;
    }
    private static void print(int[] A){
        for(int i=0;i<A.length;i++)
            System.out.print(A[i]+" ");
    }
}

```



```
}  
}
```

32.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 $\sim(\sim 0 < 4)$
- (3)将上面二者进行 & 运算。

```
import java.util.Scanner;  
public class Prog32{  
    public static void main(String[] msg){  
        //输入一个长整数  
        Scanner scan = new Scanner(System.in);  
        long l = scan.nextLong();  
        scan.close();  
        //以下截取字符  
        String str = Long.toString(l);  
        char[] ch = str.toCharArray();  
        int n = ch.length;  
        if(n<7)  
            System.out.println("输入的数小于 7 位！");  
        else  
            System.out.println("截取的 4~7 位数字：  
"+ch[n-7]+ch[n-6]+ch[n-5]+ch[n-4]);  
    }  
}
```



33.打印出杨辉三角形（要求打印出 10 行如下图）

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
```

```
public class Prog33{
    public static void main(String[] args){
        int[][] n = new int[10][21];
        n[0][10] = 1;
        for(int i=1;i<10;i++){
            for(int j=10-i;j<10+i+1;j++){
                n[i][j] = n[i-1][j-1]+n[i-1][j+1];
            }
        }
        for(int i=0;i<10;i++){
            for(int j=0;j<21;j++){
                if(n[i][j]==0)
                    System.out.print(" ");
                else{
                    if(n[i][j]<10)
                        System.out.print(" "+n[i][j]);
                    else if(n[i][j]<100)
                        System.out.print(" "+n[i][j]);
                    else
                        System.out.print(n[i][j]);
                }
            }
        }
    }
}
```



```
        System.out.println();
    }
}
}
```

34.输入 3 个数 a,b,c, 按大小顺序输出。

程序分析：利用指针方法。

```
import java.util.Scanner;
public class Prog34{
    public static void main(String[] args){
        System.out.print("请输入 3 个数: ");
        Scanner scan = new Scanner(System.in).useDelimiter("\\s");
        int a = scan.nextInt();
        int b = scan.nextInt();
        int c = scan.nextInt();
        scan.close();
        if(a<b){
            int t = a;
            a = b;
            b = t;
        }
        if(a<c){
            int t = a;
            a = c;
            c = t;
        }
        if(b<c){
            int t = b;
```



```

        b = c;
        c = t;
    }
    System.out.println(a+" "+b+" "+c);
}
}

```

35.选择排序

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。

```

import java.util.Scanner;
public class Prog35{
    public static void main(String[] args){
        System.out.print("请输入一组数: ");
        Scanner scan = new Scanner(System.in).useDelimiter("\\s");
        int[] a = new int[50];
        int m = 0;
        while(scan.hasNextInt()){
            a[m++] = scan.nextInt();
        }
        scan.close();
        int[] b = new int[m];
        for(int i=0;i<m;i++)
            b[i] = a[i];
        for(int i=0;i<b.length;i++)
            for(int j=0;j<b.length-i-1;j++)
                if(b[j]<b[j+1]){
                    int temp = b[j];
                    b[j] = b[j+1];

```




```

        b[j+1] = temp;
    }
    for(int i=0;i<b.length;i++)
        System.out.print(b[i]+" ");

}
}

```

36.交换位置

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

```

import java.util.Scanner;
public class Prog36{
    public static void main(String[] args){
        final int N = 10;
        System.out.print("请输入 10 个数的数组: ");
        Scanner scan = new Scanner(System.in);
        int[] a = new int[N];
        for(int i=0;i<a.length;i++)
            a[i] = scan.nextInt();
        System.out.print("请输入一个小于 10 的数: ");
        int m = scan.nextInt();
        scan.close();
        int[] b = new int[m];
        int[] c = new int[N-m];
        for(int i=0;i<m;i++)
            b[i] = a[i];
        for(int i=m,j=0;i<N;i++,j++)

```



```

        c[j] = a[i];
    for(int i=0;i<N-m;i++)
        a[i] = c[i];
    for(int i=N-m,j=0;i<N;i++,j++)
        a[i] = b[j];
    for(int i=0;i<a.length;i++)
        System.out.print(a[i]+" ");
    }
}

```

37.排序问题

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。

```

import java.util.Scanner;
public class Prog37{
    public static void main(String[] args){
        System.out.print("请输入一个整数: ");
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        scan.close();
        //定义数组变量标识某人是否还在圈内
        boolean[] isIn = new boolean[n];
        for(int i=0;i<isIn.length;i++)
            isIn[i] = true;
        //定义圈内人数、报数、索引
        int inCount = n;
        int countNum = 0;
        int index = 0;
    }
}

```



```

while(inCount>1){
    if(isIn[index]){
        countNum++;
        if(countNum==3){
            countNum = 0;
            isIn[index] = false;
            inCount--;
        }
    }
    index++;
    if(index==n)
        index = 0;
}
for(int i=0;i<n;i++)
    if(isIn[i])
        System.out.println("留下的是: "+(i+1));
}
}

```

38.计算字符串总长度

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。

```

import java.util.Scanner;
public class Prog38{
    public static void main(String[] args){
        System.out.print("请输入一串字符: ");
        Scanner scan = new Scanner(System.in).useDelimiter("\\n");
        String strIn = scan.next();
        scan.close();
    }
}

```



```
char[] ch = strIn.toCharArray();
System.out.println(strIn+"共"+(ch.length-1)+"个字符");
}
}
```

39.求和

编写一个函数，输入 n 为偶数时，调用函数求 $1/2+1/4+\dots+1/n$ ，当输入 n 为奇数时，调用函数 $1/1+1/3+\dots+1/n$ (利用指针函数)

```
import java.util.Scanner;
public class Prog39{
    public static void main(String[] args){
        System.out.print("请输入一个整数: ");
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        scan.close();
        if(n%2==0)
            System.out.println("结果: "+even(n));
        else
            System.out.println("结果: "+odd(n));
    }
    //奇数
    static double odd(int n){
        double sum = 0;
        for(int i=1;i<n+1;i+=2){
            sum += 1.0/i;
        }
        return sum;
    }
}
```



```
//偶数
static double even(int n){
    double sum = 0;
    for(int i=2;i<n+1;i+=2){
        sum += 1.0/i;
    }
    return sum;
}
}
```

40.字符串排序。

```
public class Prog40{
    public static void main(String[] args){
        String[] str = {"abc","cad","m","fa","f"};
        for(int i=str.length-1;i>=1;i--){
            for(int j=0;j<=i-1;j++){
                if(str[j].compareTo(str[j+1])<0){
                    String temp = str[j];
                    str[j] = str[j+1];
                    str[j+1] = temp;
                }
            }
        }
        for(String subStr:str)
            System.out.print(subStr+" ");
    }
}
```



41.递归

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？

```
public class Prog41{
    public static void main(String[] args){
        int n;
        n = fun(0);
        System.out.println("原来有"+n+"个桃子");
    }
    private static int fun(int i){
        if(i==5)
            return 1;
        else
            return fun(i+1)*5+1;
    }
}
```

42. $809??=800??+9*??+1$

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。

```
public class Prog42{
    public static void main(String[] args){
        int n = 0;
```



```
boolean flag = false;
for(int i=10;i<100;i++)
    if(809*i==800*i+9*i+1){
        flag = true;
        n = i;
        break;
    }
if(flag)
    System.out.println(n);
else
    System.out.println("无符合要求的数! ");
}
```

43.求 0—7 所能组成的奇数个数。

```
public class Prog43{
    public static void main(String[] args){
        int count = 0;
        //声明由数字组成的数
        int n = 8;
        //一位数
        count = n/2;
        //两位数
        count += (n-1)*n/2;
        //三位数
        count += (n-1)*n*n/2;
        //四位数
        count += (n-1)*n*n*n/2;
        //五位数
```



```

        count += (n-1)*n*n*n*n/2;
        //六位数
        count += (n-1)*n*n*n*n*n/2;
        //七位数
        count += (n-1)*n*n*n*n*n*n/2;
        System.out.println("0-7 所能组成的奇数个数: "+count);
    }
}

```

44.一个偶数总能表示为两个素数之和。

```

import java.util.Scanner;
public class Prog44{
    public static void main(String[] args){
        System.out.print("请输入一个偶数: ");
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        scan.close();
        if(n%2!=0){
            System.out.println("您输入的不是偶数!");
            return;
        }
        twoAdd(n);
    }
    //偶数分解为素数之和
    private static void twoAdd(int n){
        for(int i=2;i<n/2+1;i++){
            if(isPrime(i)&&isPrime(n-i)){
                System.out.println(n+"="+i+"+"+(n-i));
                break;
            }
        }
    }
}

```




```

        }
    }
}
//判断素数
private static boolean isPrime(int m){
    boolean flag = true;
    for(int i=2;i<Math.sqrt(m)+1;i++){
        if(m%i==0){
            flag = false;
            break;
        }
    }
    return flag;
}
}

```

45.判断一个素数能被几个 9 整除

```

import java.util.Scanner;
public class Prog45{
    public static void main(String[] args){
        System.out.print("请输入一个数: ");
        Scanner scan = new Scanner(System.in);
        long l = scan.nextLong();
        long n = l;
        scan.close();
        int count = 0;
        while(n>8){
            n /= 9;
            count++;
        }
    }
}

```



```

    }
    System.out.println(l+"能被"+count+"个9 整除。");
}
}
}

```

46.两个字符串连接程序

```

public class Prog46{
    public static void main(String[] args){
        String str1 = "lao lee";
        String str2 = "牛刀";
        String str = str1+str2;
        System.out.println(str);
    }
}

```

47.打印练习

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 *。

```

import java.util.Scanner;
public class Prog47{
    public static void main(String[] args){
        System.out.print("请输入 7 个整数(1-50): ");
        Scanner scan = new Scanner(System.in);
        int n1 = scan.nextInt();
        int n2 = scan.nextInt();
        int n3 = scan.nextInt();
        int n4 = scan.nextInt();
    }
}

```



```

        int n5 = scan.nextInt();
        int n6 = scan.nextInt();
        int n7 = scan.nextInt();
        scan.close();
        printStar(n1);
        printStar(n2);
        printStar(n3);
        printStar(n4);
        printStar(n5);
        printStar(n6);
        printStar(n7);
    }
    static void printStar(int m){
        System.out.println(m);
        for(int i=0;i<m;i++)
            System.out.print("*");
        System.out.println();
    }
}

```

48.加密算法

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。

```

public class Prog48{
    public static void main(String[] args){
        int n = 1234;
        int[] a = new int[4];
    }
}

```



```

        for(int i=3;i>=0;i--){
            a[i] = n%10;
            n /= 10;
        }
        for(int i=0;i<4;i++){
            System.out.print(a[i]);
            System.out.println();
            for(int i=0;i<a.length;i++){
                a[i] += 5;
                a[i] %= 10;
            }
            int temp1 = a[0];
            a[0] = a[3];
            a[3] = temp1;
            int temp2 = a[1];
            a[1] = a[2];
            a[2] = temp2;
            for(int i=0;i<a.length;i++){
                System.out.print(a[i]);
            }
        }
    }
}

```

49.计算字符串中子串出现的次数

```

public class Prog49{
    public static void main(String[] args){
        String str = "I come from County DingYuan Province AnHui.";
        char[] ch = str.toCharArray();
        int count = 0;
        for(int i=0;i<ch.length;i++){

```



```

        if(ch[i]!=' ')
            count++;
    }
    count++;
    System.out.println("共有"+count+"个字串");
}
}

```

50.求平均数

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），计算出平均成绩，将原有的数据和计算出的平均分数存放在磁盘文件"stud"中。

```

import java.io.*;
public class Prog50{
    //定义学生模型
    String[] number = new String[5];
    String[] name = new String[5];
    float[][] grade = new float[5][3];
    float[] sum = new float[5];
    public static void main(String[] args) throws Exception{
        Prog50 stud = new Prog50();
        stud.input();
        stud.output();
    }
    //输入学号、姓名、成绩
    void input() throws IOException{
        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
    }
}

```



```
//录入状态标识
boolean isRecord = true;
while(isRecord){
    try{
        for(int i=0;i<5;i++){
            System.out.print("请输入学号: ");
            number[i] = br.readLine();
            System.out.print("请输入姓名: ");
            name[i] = br.readLine();
            for(int j=0;j<3;j++){
                System.out.print("请输入第"+(j+1)+"门课成绩: ");
                grade[i][j] = Integer.parseInt(br.readLine());
            }
            System.out.println();
            sum[i] = grade[i][0]+grade[i][1]+grade[i][2];
        }
        isRecord = false;
    }catch(NumberFormatException e){
        System.out.println("请输入一个数字! ");
    }
}

//输出文件
void output() throws IOException{
    FileWriter fw = new FileWriter("E://java50//stud.txt");
    BufferedWriter bw = new BufferedWriter(fw);
    bw.write("No.  "+"Name  "+"grade1  "+"grade2  "+"grade3
"+"average");
    bw.newLine();
    for(int i=0;i<5;i++){
        bw.write(number[i]);
    }
}
```



```

        bw.write("  "+name[i]);
        for(int j=0;j<3;j++)
            bw.write("  "+grade[i][j]);
        bw.write("  "+(sum[i]/5));
        bw.newLine();
    }
    bw.close();
}
}

```

公众号：架构师专栏