0 java 程序的运行环境搭建

0.1 下载 JDK 并安装

（1）下载地址：http://www.oracle.com/technetwork/java/javase/downloads/index.html
（2）根据自己电脑操作系统的类型选择合适的版本，并下载，如下图所示：

Java SE Development Kit 8u121

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.rpm
Linux x86	177.13 MB	jdk-8u121-linux-i586.tar.gz
Linux x64	159.96 MB	jdk-8u121-linux-x64.rpm
Linux x64	174.76 MB	jdk-8u121-linux-x64.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x64.dmg
Solaris SPARC 64-bit	139.64 MB	jdk-8u121-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x64	140.42 MB	jdk-8u121-solaris-x64.tar.Z
Solaris x64	96.9 MB	jdk-8u121-solaris-x64.tar.gz
Windows x86	189.36 MB	jdk-8u121-windows-i586.exe
Windows x64	195.51 MB	jdk-8u121-windows-x64.exe

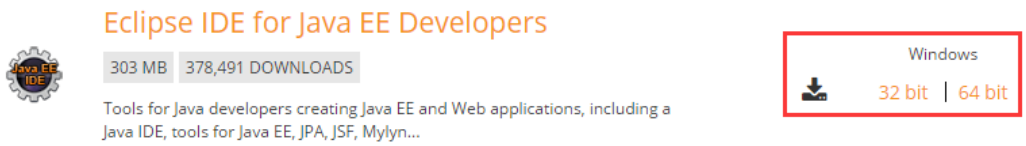
注：本书使用的 JDK 版本为：jdk1.8.0_121
（3）双击下载下来的.exe 文件，按照提示一步一步安装完成。注意：安装路径下不要存在中文。

0.2 配置环境变量

（1）环境变量的配置方法参照 www.baidu.com，此处不详述。（例如在百度中输入“win10 环境变量配置”，就可以找到对应的配置方法。）
（2）打开控制台窗口，在窗口中通过运行 java –version 和 javac 两个命令来验证环境变量是否配置正确，如下图所示：

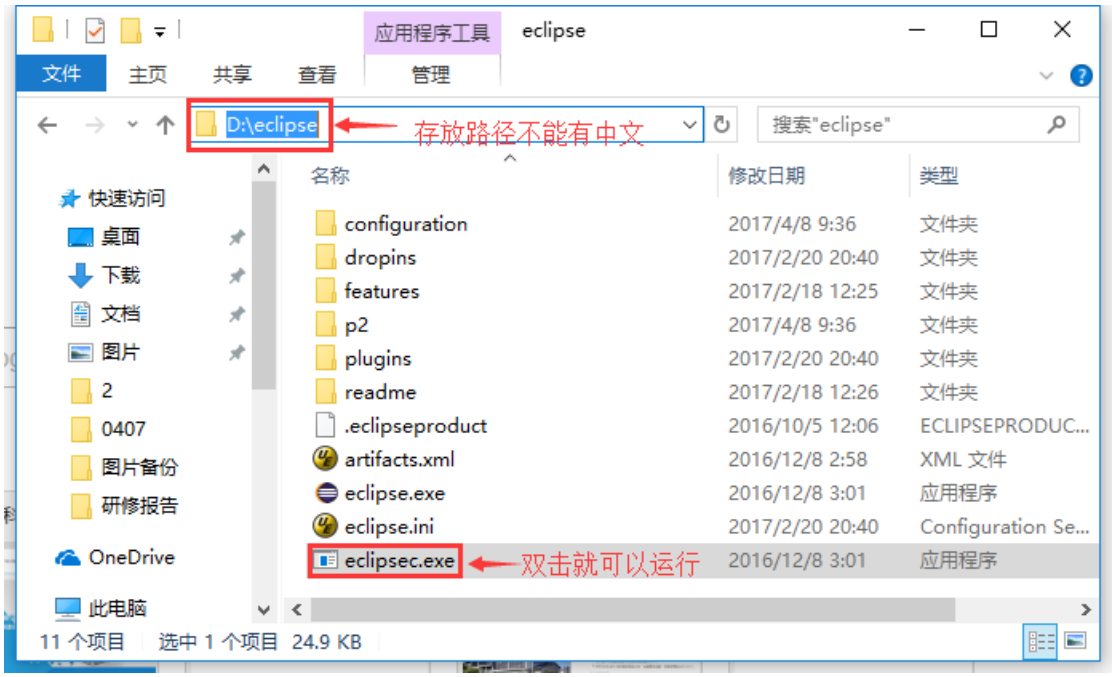
0.3 下载 eclipse

（1）下载地址：http://www.eclipse.org/downloads/eclipse-packages/，如下图所示：



根据自己电脑操作系统的类型下载对应的 eclipse 版本。注：此处下载的是 eclipse 的硬盘版文件（文件后缀名为 zip），而不是安装文件（文件后缀名为 exe），不建议大家下载安装文件。

（2）解压 eclipse 压缩包到一个合适的位置就可以使用了。（注意：eclipse 解压后的存放路径不要存在中文路径），如下图所示：



1 实验内容

实验 1 JAVA 语言基础

1. （显示五条消息）编写程序，显示 Welcome to Java 五次。
2. （打印表格）编写程序，显示以下表格：

a	a^2	a^3
1	1	1
2	4	8
3	9	27
4	16	64

3. （计算表达式）编写程序，显示以下公式的结果。

$$\frac{9.5\times4.5-2.5\times3}{45.5-3.5}$$

4. （数列求和）编写程序，显示 1+2+3+4+5+6+7+8+9 的结果。
5. （近似求 π）可以使用以下公式计算 π：

$$\pi=4\times\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\cdots\right)$$

编写程序，显示 $4\times\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}\right)$ 和 $4\times\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\frac{1}{13}\right)$ 的结果，

注意：在程序中需要使用 1.0 代替 1。

6. （圆的面积和周长）编写程序，使用以下公式计算并显示半径为 5.5 的圆的面积和周长。

周长=2*半径*π

面积=半径*半径*π

7. （以英里计的平均速度）假设一个跑步者 45 分钟 30 秒内跑了 14 公里。编写一个程序显示以每小时多少英里为单位的平均速度值。（1 英里等于 1.6 公里）
8. （人口估算）美国人口调查局基于以下假设进行人口估算：
   - 每 7 秒有一个人诞生
   - 每 13 秒有一个人死亡
   - 每 45 秒有一个移民迁人

   编写一个程序，显示未来 5 年的每年的人口数。假设当前的人口是 312032486，每年有 365 天。提示：Java 中，两个整数相除，结果还是整数，小数部分被去掉。例如，5/4 等于 1（而不是 1.25），10/4 等于 2（而不是 2.5）。如果想得到有小数部分的精确结果，进行除法运算的两个值之一必须是一个具有小数点的数值。例如，5.0/4 等于 1.25，10/4.0 等于 2.5。
9. （以公里计的平均速度）假设一个跑步者 1 小时 40 分钟 35 秒内跑了 24 英里。编写一个程序显示以每小时多少公里为单位的平均速度值。（注意，1 英里等于 1.6 公里）
10. （代数：求解 2*2 线性方程）可以使用 Cramer 规则解下面的 2*2 线性方程组：

$$ax + by = e \qquad cx + dy = f$$

$$x = \frac{ed - bf}{ad - bc} \qquad y = \frac{af - ec}{ad - bc}$$

编写程序，求解以下方程组并显示 x 和 y 的值。

$$3.4x + 50.2y = 44.5$$

$$2.1x + 0.55y = 5.9$$

11. （将摄氏溫度转换为华氏溫度）编写程序，从控制台读入 double 型的摄氏温度，然后将其转换为华氏温度，并且显示结果。转换公式如下所示：

华氏温度=（9/5）*摄氏温度+32

提示：在 Java 中，9/5 的结果是 1，但是 9.0/5 的结果是 1.83。

下面是一个运行示例：

```
Enter a temperature in Celsius: 66
66.0 Celsius is 150.8 Fahrenheit
```

12. （计算圆柱体的体积）编写程序，读入圆柱体的半径和高，并使用下列公式计算圆柱的体积

面积=半径*半径*π

体积=面积*高

下面是一个运行示例：

```
Enter the radius and length of a cylinder: 6.6 12
The area is 136.84766039999997
The volume of the cylinder is 1642.1719247999995
```

13. （将磅转换为千克）编写程序，将磅数转换为千克数。程序提示用户输入磅数，然后转换成千克并显示结果。一磅等于 0.454 千克。下面是一个运行示例：

```
Enter a number in pounds: 65.5
65.5 pounds is 29.737000000000002 kilograms
```

14. （财务应用程序：计算小费）编写一个程序，读入一笔费用与酬金率，计算酬金和总钱数。例如，如果用户输入 10 作为费用，15%作为酬金率，计算结果显示酬金为$1.5，总费用为$11.S。下面是一个运行示例：

```
Enter subtotal and gratuity rate: 10 15
The gratuity is 1.5 total is 11.5
```

15. （求一个整数各位数的和）编写程序，读取一个在 0 和 1000 之间的整数，并将该整数的各位数字相加。例如：整数是 932，各位数字之和为 14。
提示：利用操作符%分解数字，然后使用操作符/去掉分解出来的数字。例如：932%10=2，932/10=93。下面是一个运行示例：

```
Enter an integer between 0 and 1000: 12345
The sum of all digits in 12345 is 12
```

16. （求出年数）编写程序，提示用户输入分钟数（例如十亿）然后显示这些分钟代表多少年和多少天。为了简化问题，假设一年有 365 天。下面是一个运行示例：

```
Enter the number of minutes: 1000000000
1000000000 minutes is approximately 1902 years and 214 days
```

17. (当前时间）程序清单 2-7（参见教材 P_{45}）给出了显示当前格林威治时间的程序。修改这个程序，提示用户输入相对于 GMT 的时区偏移量，然后显示在这个特定时区的时间。下面是一个运行示例：

```
Enter the time zone offset to GMT: -5 Enter
The current time is 4:50:34
```

18. （物理：加速度）平均加速度定义为速度的变化量除以这个变化所用的时间，如下式所示：

$$a = \frac{v_1 - v_0}{t}$$

编写程序，提示用户输入以米/秒为单位的起始速度 V。，以米/秒为单位的终止速度 V_1，以及以秒为单位的时间段 t，最后显示平均加速度。下面是一个运行示例：

```
Enter v0, v1, and t: 5.5 50.9 4.5
The average acceleration is 10.088888888888889
```

19. （科学：计算能量）编写程序，计算将水从初始温度加热到最终温度所需的能量。程序应该提示用户输入水的重量（以千克为单位），以及水的初始温度和最终温度。计算能量的公式是：

$$Q = M \times (\text{最终温度} - \text{初始温度}) \times 4184$$

这里的 M 是以千克为单位的水的重量，温度以摄氏度为单位，而能量 Q 以焦耳为单位。下面是一个运行示例：

```
Enter the amount of water in kilograms: 55.5
Enter the initial temperature: 3.5
Enter the final temperature: 10.5
The energy needed is 1625484.0
```

20. （物理：求出跑道长度）假设一个飞机的加速度是 a 而起飞速度是 v，那么可以使用下面的公式计算出飞机起飞所需的最短跑道长度：

$$\text{跑道长度} = \frac{v^2}{2a}$$

编写程序，提示用户输入以米/秒（m/s）为单位的速度 v 和以米/秒的平方（m/s^2）为单位的加速度 a，然后显示最短跑道长度。下面是一个运行示例：

```
Enter speed v: 60
Enter acceleration a: 3.5
The minimum runway length for this airplane is 514.2857142857143 meters
```

21. （财务应用程序：复利值）假设你每月向银行账户存 100 美元，年利率为 5%，那么每月利率是 0.05/12=0.00417。第一个月之后，账户上的值就变成：

100*（1+0.00417）=100.417

第二个月之后，账户上的值就变成：

（100+100.417）*（1+0.00417）-201.252

第三个月之后，账户上的值就变成：

（100+201.252）*（1+0.00417）=302.507

依此类推。

编写程序提示用户输入一个数目（例如：100）、年利率（例如：5）以及月份数（例如：6），然后显示给定月份后账户上的钱数。下面是一个运行示例：

```
Enter the amount to be saved for each month: 100
Enter the annual interest rate: 5
Enter the number of months: 6
After the 6th month, the account value is 608.811017705596
```

22. （医疗应用程序：计算 BMI）身体质量指数（BMI）是对体重的健康测量。它的值可以通过将体重（以公斤为单位）除以身高（以米为单位）的平方值得到。编写程序，提示用户输入体重（以磅为单位）以及身高（以英寸为单位），然后显示 BMI。注意：一磅是 0.4S359237 公斤，一英寸是 0.0254 米。下面是一个运行示例：

```
Enter weight in pounds: 95.5
Enter height in inches: 50
BMI is 26.857257942215885
```

23. （几何：两点间距离）编写程序，提示用户输入两个点（xl，yl）和（x2，y2），然后显示两点间的距离。计算两点间距离的公式是$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$。注意：可以使用 Math.pow（a，0.5）来计算$\sqrt{a}$。下面是一个运行示例：

```
Enter x1 and y1: 1.5 3.4
Enter x2 and y2: 4 5
The distance of the two points is 2.968164415931166
```

24. （几何：六边形面积）编写程序，提示用户输入六边形的边长，然后显示它的面积。计算六边形面积的公式是：

$$面积=\frac{3\sqrt{3}}{2}s^2$$

这里的 s 就是边长。下面是一个运行示例:

```
Enter the side: 6
The area of the hexagon is 93.52799999999999
```

25. （科学：风寒温度）外面到底有多冷？只有温度是不足以提供答案的，包括风速、相对湿度以及阳光等其他的因索在确定室外是否寒冷方面都起了很重要的作用。2001年，国家气象服务（NWS）利用温度和风速计算新的风寒温度，来衡量寒冷程度。计算公式如下所示：

$$t_{wc}=35.74+0.6215t_a-35.75v^{0.16}+0.4275t_av^{0.16}$$

这里的 t_a 是室外的温度，以华氏摄氏度为单位，而 v 是速度，以每小时英里数为单位。

t_{wc} 是风寒温度。该公式不适用于风速低于 2mph，或温度在-58°F 以下或 41°F 以上的情况。编写程序，提示用户输入在-58°F 和 41°F 之间的度数，同时大于或等于 2 的风速，然后显示风寒温度。使用 Math.pow（a.b）来计算 $v^{0.16}$。下面是一个运行示例：

```
Enter the temperature in Fahrenheit between -58 and 41: 5.3
Enter the wind speed miles per hour (must be greater than or equal to 2) : 6
The wind chill index is -5.567068455881625
```

26. （打印表格）编写程序，显示下面的表格。将浮点数值类型转换为整数。

a	b	pow(a, b)
1	2	1
2	3	8
3	4	81
4	5	1024
5	6	15625

27. （几何：三角形的面积）编写程序，提示用户输入三角形的三个点（xl，yl）、（x2，y2）和（x3.y3），然后显示它的面积。计算三角形面积的公式是：

$$s = (\text{边}1 + \text{边}2 + \text{边}3)/2$$

$$\text{面积} = \sqrt{s(s-\text{边}1)(s-\text{边}2)(s-\text{边}3)}$$

下面是一个运行示例

```
Enter three points for a triangle: 1.5 -3.4 4.6 5 9.5 -3.4
The area of the triangle is 33.600000000000016
```

28. （财务应用程序：计算利息）如果知道收支余额和年利率的百分比，就可以使用下面的公式计算

$$\text{利息额} - \text{收支余额} \times (\text{年利率} / 1200)$$

编写程序，读取收支余额和年百分利率，显示下月利息。下面是一个运行示例：

```
Enter balance and annual interest rate: 1000 3.5
The interest is 2.91
```

29. （财务应用：计算未来投资值）编写程序，读取投资总额、年利率和年数，然后使用下面的公式显示未来投资金额：

未来投资金额=投资总额*（1+月利率）

例如：如果输入的投资金额为 1000，年利率为 3.25%，年数为 1，那么未来投资额为 1032.98。下面是一个运行示例：

```
Enter the investment amount, for example 120000.95: 1000.56
Enter annual interest rate, for example 8.25: 4.25
Enter number of years as an integer, for example 5: 1
Future value is 1043.92
```

30. （驾驶费用）编写一个程序，提示用户输入驾驶的距离、以每加仑多少英里的汽车燃油性能，以及每加仑的价格，然后显示旅程的费用。下面是一个运行示例：

```
Enter the driving distance: 900.5
Enter miles per gallon: 25.5
Enter price per gallon: 3.55
The cost of driving is $125.36372549019607
```

31. （代数：解一元二次方程）可以使用下面的公式求一元二次方程 $ax^2+bx+c=0$ 的两个根：

$$r_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \text{ 和 } r_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

b^2-4ac 称作一元二次方程的判别式。如果它是正值，那么一元二次方程就有两个实数根。如果它为 0，方程式就只有一个根。如果它是负值，方程式无实根。

编写程序，提示用户输入 a、b 和 c 的值，并且显示基于判别式的结果。如果这个判别式为正，显示两个根。如果判别式为 0，显示一个根。否则，显示“Theequationhasnorealroots”（该方程式无实数根）。

注意：可以使用 Math.pow（x，0.5）来计算$\sqrt{x}$，下面是一些运行示例。

Entera，b，c:1.031

Theequationhastworoots-0.381966and-2.61803

Entera，b，c:12.01

Theequationhasoneroot-1

Enteratbtc:123

Theequationhasnorealroots

32. （游戏：三个数的加法）编写程序，随机产生三个一位整数.并提示用户输入这三个整数的和，判断用户输入的和是否正确。
33. （随机月份）编写一个随机产生 1 和 12 之间整数的程序，并且根据数字 1，2，3，.....，12 显示相应的英文月份：January，February，…，December。
34. （找到将来的日期）编写一个程序，提示用户输入代表今天日期的数字（周日为 0，周一为 1，……，周六为 6）。同时，提示用户输入一个今天之后的天数，作为代表将来某天的数字，然后显示这天是星期几。下面是一个运行示例：

Entertoday'sday:1

Enterthenumberofdayselapsedsincetoday:3

TodayisMondayandthefuturedayisThursday

Entertoday'sday:0

Enterthenumberofdayselapsedsincetoday:31

TodayisSundayandthefuturedayisWednesday

35. （医疗应用程序：BMI）修改教材程序清单 3-4（参见教材 P76），让用户输入重量、英尺和英寸。例如：一个人身高是 5 英尺 10 英寸，输入的英尺值就是 5、英寸值为 10。下面是一个运行示例：

```
Enter weight in pounds: 140 Enter
Enter feet: 5 Enter
Enter inches: 10 Enter
BMI is 20.087702275404553
Normal
```

36. （对三个整数排序）编写程序，提示用户输入三个整数。以非降序的形式显示这三个整数。
37. （商业：检查 ISBN-10）ISBN-10（国际标准书号）以前是一个 10 位整数 $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}$，最后的一位 d_{10} 是校验和，它是使用下面的公式用另外 9 个数计算出来的：

$$(d_1 \times 1 + d_2 \times 2 + d_3 \times 3 + d_4 \times 4 + d_5 \times 5 + d_6 \times 6 + d_7 \times 7 + d_8 \times 8 + d_9 \times 9) \% 11$$

如果校验和为 10，那么按照 1SBN-10 的习惯，最后一位应该表示为 X。编写程序，提示用户输入前 9 个数，然后显示 10 位 ISBN（包括前面起始位置的 0）。程序应该读取一个整数输入。以下是运行示例：

```
Enter the first 9 digits of an ISBN as integer: 013601267
The ISBN-10 number is 0136012671
```

```
Enter the first 9 digits of an ISBN as integer: 013031997
The ISBN-10 number is 013031997X
```

38. （游戏：加法测验）教材程序清单 3-3（参见教材 P75）随机产生一个减法问题。修改这个程序，随机产生一个计算两个小于 100 的整数的加法问题。

39. （给出一个月的总天数）编写程序，提示用户输入月份和年份，然后显示这个月的天数。例如：如果用户输入的月份是 2 而年份是 2012，那么程序应该显示“February 2012 has 29 days”（2012 年 2 月有 29 天）。如果用户输入的月份为 3 而年份为 2015，那么程序就应该显示“March 2015 has 31 days”（2015 年 3 月有 31 天）。

40. （回文数字）编写一个程序，提示用户输入一个三位的整数，然后确定它是否回文数字。当从左到右，以及从右到左都是一样的话，这个数字称为回文数。下面是程序的一个运行示例：

    Enter a three-digit integer: 121

    121 is a palindrome

    Enter a three-digit integer: 123

    123 is not a palindrome

41. （财务应用程序：计算税款）教材程序清单 3-5（参见教材 P78）给出了计算单身登记人税款的源代码。将程序清单 3-5（参见教材 P78）补充完整，从而计算所有登记的婚姻状态的税款。

42. （游戏：猜硬币的正反面）编写程序，让用户猜一猜是硬币的正面还是反面。这个程序随机产生一个整数 0 或者 1，它们分别表示硬币的正面和反面。程序提示用户输入一个猜测值，然后报告这个猜测值是正确的还是错误的。

43. （游戏：彩票）修改程序清单 3-8（参见教材 P84），产生三位整数的彩票。程序提示用户输入一个三位整数，然后依照下面的规则判定用户是否赢得奖金：

    1）如果用户输入的所有数匹配彩票的确切顺序，奖金是 10000 美元。

    2）如果用户输入的所有数匹配彩票的所有数字，奖金是 3000 美元。

    3）如果用户输入的其中一个数匹配彩票号码中的一个数，奖金是 1000 美元。

44. （随机点）编写程序，显示矩形中一个随机点的坐标。矩形中心位于（0，0）、宽 100、高 200。

45. （游戏：剪刀、石头、布）编写可以玩流行的剪刀-石头-布游戏的程序。（剪刀可以剪布，石头可以砸剪刀，而布可以包石头。）程序提示用户随机产生一个数，这个数为 0、1 或者 2，分别表示石头、剪刀和布。程序提示用户输入值 0、1 或者 2，然后显示一条消息，表明用户和计算机谁赢了游戏，谁输了游戏，或是打成平手。下面是运行示例：

    scissor（0），rock（1），paper（2）：1

    The computer is scissor. You are rock. Youwon

    scissor（0），rock（1），paper（2）：2

    The computer is paper. You are paper too. It is a draw

46. （运输成本）一个运输公司使用下面的函数，根据运输重量（以磅为单位）来计算运输成本（以美元计算）。

$$c(w)=\begin{cases}3.5, 若\ 0<w\leqslant 1\\ 5.5, 若\ 1<w\leqslant 3\\ 8.5, 若\ 3<w\leqslant 10\\ 10.5, 若\ 10<w\leqslant 20\end{cases}$$

    编写一个程序，提示用户输入包裹重量，显示运输成本。如果重量大于 20，显示一条消

息“the package can not be shipped”。

47. （计算三角形的周长）编写程序，读取三角形的三条边，如果输入值合法就计算这个三角形的周长；否则，显示这些输入值不合法。如果任意两条边的和大于第三边，那么输入值都是合法的。

48. （科学：某天是星期几）泽勒一致性是由克里斯汀•泽勒开发的用于计算某天是星期几的算法。这个公式是：

$$h=\left(q+\frac{26(m+1)}{10}+k+\frac{k}{4}+\frac{j}{4}+5j\right)\%7$$

其中：

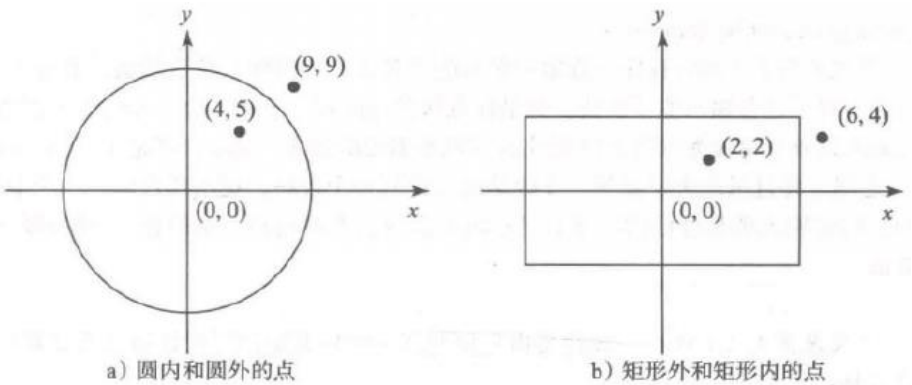
- h 是一个星期中的某一天（0 为星期六；1 为星期天；2 为星期一；3 为星期二；4 为星期三；5 为星期四；6 为星期五）。
- q 是某月的第几天。
- m 是月份（3 为三月，4 为四月，……，12 为十二月）。一月和二月分别记为上一年的 13 和 14 月。
- j 是世纪数（即 $\left\lfloor\frac{year}{100}\right\rfloor$）。
- k 是该世纪的第几年（year%100）。

注意，公式中的除法执行一个整数相除。编写程序，提示用户输入年、月和该月的哪一天，然后显示它是一周中的星期几。下面是一些运行示例：

Enter year:（e.g.，2012）:2015
Enter month: 1-12: 1
Enter the day of the month: 1-31: 25
Day of the week is Sunday
Enter year:（e.g.，2012）:2012
Enter month: 1-12: 5
Enter the day of the month: 1-31: 12
Day of the week is Saturday

提示：一月和二月在这个公式里是用 13 和 14 表示的。所以需要将用户输入的月份 1 转换为 13，将用户输入的月份 2 转换为 14，同时将年份改为前一年。

49. （几何：点是否在圆内？）编写程序，提示用户输入一个点（x，y），然后检查这个点是否在以原点（0，0）为圆心、半径为 10 的圆内。例如：（4，S）是圆内的一点，而（9，9）是圆外的一点，如下图 a 所示。



a）圆内和圆外的点　　b）矩形外和矩形内的点

提示：如果一个点到（0，0）的距离小于或等于 10，那么该点就在圆内，计算距离的公

式是$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$，使用各种情况来测试你的程序。以下是两个运行示例。

Enter a point with two coordinates: 4 5

Point（4.0，5.0）is in the circle

Enter a point with two coordinates: 9 9

Point（9.0，9.0）is not in the circle

50. （几何：点是否在矩形内）编写程序，提示用户输入点（x，y），然后检测该点是否在以原点（0，0）为中心、宽为10、高为5的矩形中。例如：（2，2）在矩形内，而（6，4）在矩形外，如上图b所示。

提示：如果一个点到点（0，0）的水平距离小于等于10/2且到点（0，0）的垂直距离小于等于5.0/2，该点就在矩形内，使用各种情况来测试你的程序。这里有两个运行示例：

Enter a point with two coordinates: 2 2

Point（2.0，2.0）is in ther ectangle

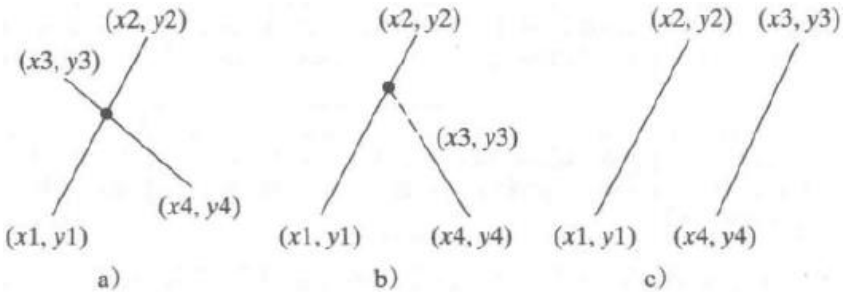
Enter a point with two coordinates: 6 4

Point（6.0，4.0）is not in the rectangle

51. （游戏：挑一张牌）编写程序，模拟从一副52张的牌中选择一张牌。程序应该显示牌的大小（Ace、2、3、4、5、6、7、8、9、10、Jack、Queen、King）以及牌的花色（Clubs（黑梅花）、Diamonds（红方块）、Hearts（红心）、Spades（黑桃））。下面是这个程序的运行示例：

The card you picked is Jack of Hearts

52. （几何：交点）第一条直线上面的两个点是（xl，yl）和（x2，y2），第二条直线的两个点是（x3，y3）和（x4，y4），如下图所示。



两条直线的交点可以通过下面的线性方程组求解：

$$(y_1-y_2)x-(x_1-x_2)y=(y_1-y_2)x_1-(x_1-x_2)y_1$$
$$(y_3-y_4)x-(x_3-x_4)y=(y_3-y_4)x_3-(x_3-x_4)y_3$$

这个线性方程组可以应用Cramer规则求解（见编程练习题13）。如果方程无解，则两条直线平行。编写一个程序，提示用户输入这四个点，然后显示它们的交点。下面是这个程序的运行示例：

Enter xl，yl，x2，y2，x3，y3，x4，y4: 2 2 5 -1.0 4.0 2.0 -1.0 -2.0

The intersecting point is at（2.88889，1.1111）

Enterx l，yl，x2，y2，x3，y3，x4，y4: 2 2 7 6.0 4.0 2.0 -1.0 -2.0f

The two lines are parallel

53. （使用操作符&&、‖和∧）编写一个程序，提示用户输入一个整数值，然后判定它是否能被5和6整除，是否能被5或6整除，以及是否能被5或6整除但是不能同时被它们整除。下面是这个程序的运行示例：

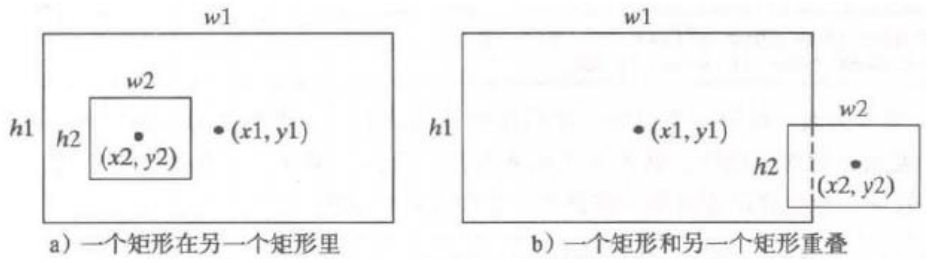
Enter an integer: 10

Is 10 divisible by 5 and 6 ? false

Is 10 divisible by 5 or 6 ? true

Is 10 divisible by 5 or 6， but not both ? true

54. （几何：两个矩形）编写一个程序，提示用户输入两个矩形中点的 x 坐标和 y 坐标以及它们的宽度和高度，然后判定第二个矩形是在第一个矩形内，还是和第一个矩形重叠，如下图所示。



a）一个矩形在另一个矩形里　　b）一个矩形和另一个矩形重叠

下面是运行示例:

Enter rl's center x-， y-coordinates， width， and height: 2.5 4 2.5 43

Enter r2's center x-， y-coordinates， width， andheight:1.5 5 0.5 3

r2 is inside rl

Enter rl's center x-， y-coordinates， width， and height: 1 2 3 5.5

Enterr2'scenterx-， y-coordinates， width， and height : 3 4 4.5 5

r2 overlaps rl

Enter rl's center x-， y-coordinates， width， and height: 1 2 3 3

Enter r2's center x-， y-coordinates， width， andheight: 40 45 3 2

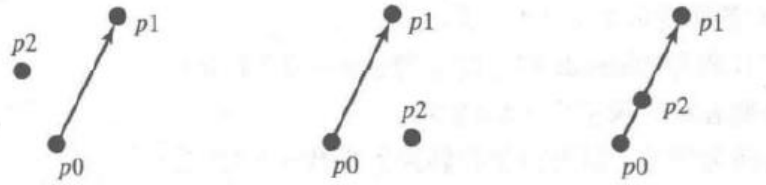
r2 does not overlap rl

55. （金融：货币兑换）编写一个程序，提示用户输入从美元到人民币的兑换汇率。提示用户输入 0 表示从美元兑换为人民币，输入 1 表示从人民币兑换为美元。继而提示用户输入美元数量或者人民币数量，分别兑换为另外一种货币。下面是运行示例：

```
Enter the exchange rate from dollars to RMB: 6.81
Enter 0 to convert dollars to RMB and 1 vice versa: 0
Enter the dollar amount: 100
$100.0 is 681.0 Yuan
Enter the exchange rate from dollars to RMB: 6.81
Enter 0 to convert dollars to RMB and 1 vice versa: 1
Enter the RMB amount: 10000
10000.0 Yuan is $1468.4287812041116
Enter the exchange rate from dollars to RMB: 6.81
Enter 0 to convert dollars to RMB and 1 vice versa: 5
Incorrect input
```

56. （几何：点的位置）给定一个从点 p0（x0，y0）到 pl（xl，pl）的有向线段，可以使用下面的条件来确定点 p2（x2，y2）是在线段的左侧、右侧，或者在该直线上（见下图）：

$$(x1-x0)\times(y2-y0)-(x2-x0)\times(y1-y0)\begin{cases}>0 & p2\text{ 在线段的左侧}\\ =0 & p2\text{ 在该线段上}\\ <0 & p2\text{ 在线段的右侧}\end{cases}$$



a）p2 在线段的左侧　　b）p2 在线段的右侧　　C）p2 在该线段上

编写一个程序，提示用户输入三个点 p0、p1 和 p2，显示 p2 是否在从 p0 到 p1 的线段左侧、右侧，或者在该直线上。下面是运行示例：

```
Enter three points for p0, p1, and p2: 4.4 2 6.5 9.5 -5 4
(-5.0, 4.0) is on the left side of the line from (4.4, 2.0) to (6.5, 9.5)

Enter three points for p0, p1, and p2: 1 1 5 5 2 2
(2.0, 2.0) is on the line from (1.0, 1.0) to (5.0, 5.0)

Enter three points for p0, p1, and p2: 3.4 2 6.5 9.5 5 2.5
(5.0, 2.5) is on the right side of the line from (3.4, 2.0) to (6.5, 9.5)
```

实验 2 JAVA 语言基础(2)-数学函数、字符和字符串、循环

1. （几何：最大圆距离）最大圆距离是指球面上两个点之间的距离。假设（xl，y1）和（x2，y2）是两个点的地理经纬度。两个点之间的最大圆距离可以使用以下公式计算：

$$d = 半径 \times \arccos(\sin(x_1) \times \sin(x_2) + \cos(x_1) \times \cos(x_2) \times \cos(y_1 - y_2))$$

编写一个程序，提示用户以度为单位输入地球上两个点的经纬度，显示其最大圆距离值。地球的平均半径为 6371.01km。注意，你需要使用 Math.toRadians 方法将度转换为弧度值。公式中的经纬度是相对北边和西边的，使用负数表示相对南边和东边的度数。下面是一个运行示例：

```
Enter point 1 (latitude and longitude) in degrees: 39.55 -116.25
Enter point 2 (latitude and longitude) in degrees: 41.5 87.37
The distance between the two points is 10691.79183231593 km
```

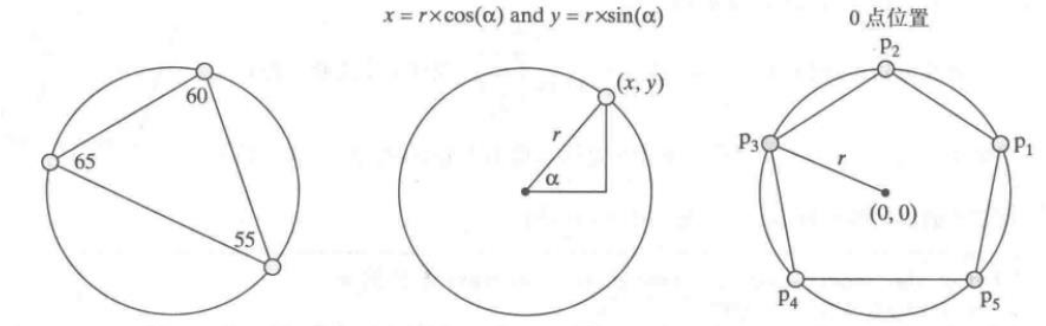
2. （几何：正多边形的面积）正多边形是一个 n 条边的多边形，它每条边的长度都相等，而且所有角的度数也相等（即多边形既等边又等角）。计算正多边形面积的公式是：

$$面积 = \frac{n \times s}{4 \times \tan\left(\quad\right)}$$

这里，s 是边长。编写一个程序，提示用户输入边的个数以及正多边形的边长，然后显示它的面积。这里是一个运行示例：

```
Enter the number of sides: 5
Enter the side: 6.5
The area of the polygon is 72.69017017488385
```

3. （圆上的随机点）编写一个程序，产生一个圆心在（0，0）、半径为 40 的圆上面的三个随机点，显示由这三个随机点组成的三角形的三个角的度数，如下图所示。（提示：产生 0~π 之间的一个以弧度为单位的随机角度 α，如下图所示，则由这个角度所确定的点为（r*cos(α)，r*sin(α)））。



a）由圆上三个随机点构成的三角形　b）可以从一个随机角度 α 产生圆上的随机点　c）一个正五边形，其中心位于 (0,0)，其中一个点位于 0 点位置

4. （给出 ASCII 码对应的字符）编写一个程序，得到一个 ASCII 码的输入（0~127 之间的一个整数），然后显示该字符。下面是一个运行示例：

```
Enter an ASCII code: 69
The character for ASCII code 69 is E
```

5. （给出字符的 Unicode 码）编写一个程序，得到一个字符的输入，然后显示其 Unicode 值。下面是一个运行示例：

```
Enter a character: t
The Unicode for the character t is 116
```

6. （十六进制转二进制）编写一个程序，提示用户输入一个十六进制数，显示其对应的二进制数。下面是几个运行示例：

```
Enter a hex character: B
The binary value is 1011

Enter a hex character: G
G is an invalid input
```

7. （转换字母等级为数字）编写一个程序，提示用户输入一个字母等级 A、B、C、D 或者 F，显示对应的数字值 4、3、2、1 或者 0。下面是运行示例：

```
Enter a letter grade: B
The numeric value for grade B is 3

Enter a letter grade: T
T is an invalid grade
```

8. （电话键盘）电话上的国际标准字母/数字映射如下图所示：

编写一个程序，提示用户输入一个字母，然后显示对应的数字。

```
Enter an uppercase letter: A
The corresponding number is 2

Enter an uppercase letter: a
The corresponding number is 2

Enter an uppercase letter: +
+ is an invalid input
```

9. （一个月中的日期）编写一个程序，提示用户输入一个年份和一个月份名称的前三个字母（第一个字母使用大写形式），显示该月中的天数。下面是运行示例：

```
Enter a year: 2001
Enter a month (first three letters with the first letter in uppercase): Jan
Jan 2001 has 31 days

Enter a year: 2016
Enter a month (first three letters with the first letter in uppercase): Feb
Feb 2016 has 29 days
```

10. （检查 SSN）编写一个程序，提示用户输入一个社保号码，它的格式是 DDD-DD-DDDD，其中 D 是一个数字。你的程序应该判断输入是否合法。下面是运行示例：

```
Enter a SSN: 232-23-5435
232-23-5435 is a valid social security number
```

```
Enter a SSN: 23-23-5435
23-23-5435 is an invalid social security number
```

11. （检测子串）编写一个程序，提示用户输入两个字符串，检测第二个字符串是否是第一个字符串的子串。

```
Enter string s1: ABCD
Enter string s2: BC
BC is a substring of ABCD

Enter string s1: ABCD
Enter string s2: BDC
BDC is not a substring of ABCD
```

12. （财务应用：酬金）编写一个程序，读取下面的信息，然后输出一个酬金声明：
雇员姓名（如：Smith）
每周的工作小时数（如，10 小时）
每小时的酬金（如，9.75 美元）
联邦所得税税率（如，20%）
州所得税税率（如，9%）
下面是一个运行示例：

```
Enter employee's name: Smith
Enter number of hours worked in a week: 10
Enter hourly pay rate: 9.75
Enter federal tax withholding rate: 0.2
Enter state tax withholding rate: 0.09
Employee Name: Smith

Hours Worked:  10.0
Pay Rate:  $9.75
Gross Pay:  $97.5
Deductions:
  Federal Withholding (20.0%):  $19.5
  State Withholding (9.0%):  $8.77
  Total Deduction:  $28.27
Net Pay:   $69.22
```

13. （对三个城市排序）编写一个程序，提示用户输入三个城市名称，然后以升序进行显示。
下面是一个运行示例：

```
Enter the first city: Chicago
Enter the second city: Los Angeles
Enter the third city: Atlanta
The three cities in alphabetical order are Atlanta Chicago Los Angeles
```

14. （生成车牌号码）假设一个车牌号码由三个大写字母和后面的四个数字组成。编写一个程序，生成一个车牌号码。.

15. （将千克转换成磅）编写程序，显示下面的表格（注意：1 千克为 2.2 磅）。

千克	磅
1	2.2
3	6.6
...	
197	433.4
199	437.8

16. （将英里转换成千米）编写程序，显示下面的表格（注意：1 英里为 1.609 千米）。

英里	千米
1	1.609
2	3.218
...	
9	14.481
10	16.090

17. （英里与千米之间的互换）编写一个程序，并排显示下列两个表格。

英里	千米	千米	英里
1	1.609	20	12.430
2	3.218	25	15.538
...			
9	14.481	60	37.290
10	16.090	65	40.398

18. （找出最高分）编写程序，提示用户输入学生的个数、每个学生的名字及其分数，最后显示得最高分的学生的名字。
19. （找出两个分数最高的学生）编写程序，提示用户输入学生的个数、每个学生的名字及其分数，最后显示获得最高分的学生和第二高分的学生。
20. （显示 ACSII 码字符表）编写一个程序，打印 ASCII 字符表从“!”到“~”的字符。每行打印 10 个字符。ASCII 码表参见教材附录。数字之间用一个空格字符隔开。
21. （找出一个整数的因子）编写程序，读入一个整数，然后以升序显示它的所有最小因子。例如，若输入的整数是 120，那么输出就应该是：2，2，2，3，5。
22. （打印金字塔形的数字）编写一个嵌套的 for 循环，打印下面的输出

```
                    1
                  1   2   1
                1   2   4   2   1
              1   2   4   8   4   2   1
            1   2   4   8  16   8   4   2   1
          1   2   4   8  16  32  16   8   4   2   1
        1   2   4   8  16  32  64  32  16   8   4   2   1
      1   2   4   8  16  32  64 128  64  32  16   8   4   2   1
```

23. （财务应用程序：显示分期还贷时间表）对于给定的贷款额的月支付额包括偿还本金及利息。月利息是通过月利率乘以余额（剩余本金）计算出来的。因此，每月偿还的本金等于月支付额减去月利息。编写一个程序，让用户输入贷款总额、贷款年数以及利率，然后显示分期还贷时间表。下面是一个运行示例：

```
Enter loan amount, for example 120000.95: 10000
Enter number of years as an integer, for example 5: 1
Enter yearly interest rate, for example 8.25: 7
Monthly Payment: 865.26
Total Payment: 10383.2

Payment#        Interest        Principal       Balance
1               58.33           806.93          9193.07
2               53.62           811.64          8381.43
3               48.89           816.37          7565.06
4               44.12           821.14          6743.92
5               39.33           825.93          5917.99
6               34.52           830.74          5087.25
7               29.67           835.59          4251.66
8               24.8            840.46          3411.2
9               19.89           845.37          2565.83
10              14.96           850.3           1715.53
11              10.0            855.26          860.27
12              5.01            860.25          0.01
```

注意：最后一次偿还后，余额可能不为 0。如果是这样的话，最后一个月支付额应当是正常的月支付额加上最后的余额。

提示：编写一个循环来打印该表。由于每个月的还贷额都是相同的，因此，应当在循环之前计算它。开始时，余额就是贷款总额。在循环的每次迭代中，计算利息及本金，然后更新余额。这个循环可能会是这样的：

```
for (i = 1; i <= numberOfYears * 12; i++) {
  interest = monthlyInterestRate * balance;
  principal = monthlyPayment - interest;
  balance = balance - principal;
  System.out.println(i + "\t\t" + interest
    + "\t\t" + principal + "\t\t" + balance);
}
```

24. （数列求和）编写程序，计算下面数列的和：

$$\frac{1}{3}+\frac{3}{5}+\frac{5}{7}+\frac{7}{9}+\frac{9}{11}+\frac{11}{13}+\cdots+\frac{95}{97}+\frac{97}{99}$$

25. （计算 π）使用下面的数列可以近似计算 π：

$$\pi=4\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\cdots+\frac{(-1)^{i+1}}{2i-1}\right)$$

编写程序，显示当 i=10000，20000，……，100000 时 π 的值。

26. （显示每月第一天是星期几）编写程序，提示用户输入年份和代表该年第一天是星期几的数字，然后在控制台上显示该年每月第一天的星期。例如，如果用户输入的年份是 2013 和代表 2013 年 1 月 1 日为星期二的 2，程序应该显示如下输出：

```
January 1, 2013 is Tuesday
...
December 1, 2013 is Sunday
```

27. （显示日历）编写程序，提示用户输入年份和代表该年第一天是星期几的数字，然后在控制台上显示该年的日历表。例如，如果用户输入年份 2013 和代表 2013 年 1 月 1 日为星期二的 2，程序应该显示该年每个月的日历，如下所示：

January 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

December 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

28. （完全数）如果一个正整数等于除它本身之外其他所有除数之和，就称之为完全数。例如：6 是第一个完全数，因为 6=1+2+3。下一个完全数是 28=14+7+4+2+1。10000 以下的完全数有四个。编写程序，找出这四个完全数。
29. （游戏：石头、剪刀、布）编程练习题 50 给出玩石头-剪刀-布游戏的程序。修改这个程序，让用户可以连续地玩这个游戏，直到用户或者计算机赢对手两次以上为止。
30. （十进制到八进制）编写程序，提示用户输入一个十进制整数，然后显示对应的八进制值。在这个程序中不要使用 Java 的 Interger.toOctalString(int)方法。
31. （财务应用程序：求出销售总额）假设你已经在某百货商店开始销售工作。你的工资包括基本工资和提成。基本工资是 5000 美元。使用下面的方案确定你的提成率。

销售额	提成率
0.01 ～ 5000 美元	8%
5000.01 ～ 10 000 美元	10%
10 000.01 及以上	12%

注意：这是一个渐进税率。第一个 5000 美元的税率是 8%，下一个 5000 美元是 10%，余下的是 12%。如果销售额是 25000，提成则为 5000*8%+5000*10%+15000*12%=2700。你的目标是一年挣 30 000 美元。编写程序找出为挣到 30 000 美元，你所必须完成的最小销售额。

32. （模拟：正面或反面）编写程序，模拟抛硬币一百万次.显示出现正面和反面的次数。
33. （最大数的出现次数）编写程序读取整数，找出它们的最大数，然后计算该数的出现次数。假设输入是以 0 结束的。假定输入是 3 5 2 5 5 5 0，程序找出最大数 5，而 5 出现的次数是 4。

提示：维护 max 和 count 两个变量。max 存储当前最大数，而 count 存储它的出现次数。

初始状态时，将第一个数赋值给 max 而将 count 赋值为 1。然后将接下来的每个数字逐个地和 max 进行比较。如果这个数大于 max，就将它赋值给 max，同时将 count 重置为 1。如果这个数等于 max，就给 count 加 1。

```
Enter numbers: 3 5 2 5 5 5 0 Enter
The largest number is 5
The occurrence count of the largest number is 4
```

34. （统计：计算平均值和标准方差）在商务应用程序中经常需要计算数据的平均值和标准方差。平均值就是数字的简单平均。标准方差则是一个统计数字，给出了在一个数字集中各种数据到底离开平均值的聚集度有多紧密。例如，一个班级的学生的平均年龄是多少？年龄相差近吗？如果所有的学生都是同龄的，那么方差为 0。编写一个程序，提示用户输入 10 个数字，然后运用下面的公式，显示这些数字的平均数以及标准方差。

平均值=-^

$$\text{平均值} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n} \qquad \text{方差} = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}}{n-1}}$$

下面是一个运行示例：

```
Enter ten numbers: 1 2 3 4.5 5.6 6 7 8 9 10 Enter
The mean is 5.61
The standard deviation is 2.99794
```

35. （商业：检测 ISBN-13）ISBN-13 是一个标识书籍的新标准。它使用 13 位数字 $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}d_{11}d_{12}d_{13}$。最后一位数字 d_{13} 是一个校验和，是使用下面的公式从其他数字中计算出来的：

$$10 - (d_1 + 3d_2 + d_3 + 3d_4 + d_5 + 3d_6 + d_7 + 3d_8 + d_9 + 3d_{10} + d_{11} + 3d_{12}) \ \%10$$

如果校验和为 10，将其替换为 0。程序应该将输入作为一个字符串读入。下面是运行示例：

```
Enter the first 12 digits of an ISBN-13 as a string: 978013213080 Enter
The ISBN-13 number is 9780132130806
```

```
Enter the first 12 digits of an ISBN-13 as a string: 978013213079 Enter
The ISBN-13 number is 9780132130790
```

```
Enter the first 12 digits of an ISBN-13 as a string: 97801320 Enter
97801320 is an invalid input
```

36. （最长的共同前缀）编写一个程序，提示用户输入两个字符串，显示两个字符串最长的共同前缀。下面是运行示例：

```
Enter the first string: Welcome to C++ Enter
Enter the second string: Welcome to programming Enter
The common prefix is Welcome to
```

```
Enter the first string: Atlanta Enter
Enter the second string: Macon Enter
Atlanta and Macon have no common prefix
```

实验 3 JAVA 语言基础(3)-方法、数组

1. （检测密码）一些网站对于密码具有一些规则。编写一个方法，检测字符串是否是一个有效密码。假定密码规则如下：
   - 密码必须至少 8 位字符。
   - 密码仅能包含字母和数字。
   - 密码必须包含至少两个数字。

   编写一个程序，提示用户输入一个密码，如果符合规则，则显示 Valid Password，否则显示 Invalid Password。
2. （数学：平方根的近似求法）有几种实现 Math 类中 sqrt 方法的技术。其中一个称为巴比伦法。它通过使用下面公式的反复计算近似地得到：

   ```
   nextGuess = (lastGuess + n / lastGuess) / 2
   ```

   当 nextGuess 和 lastGuess 几乎相同时，nextGuess 就是平方根的近似值。最初的猜测值可以是任意一个正值（例如 1）。这个值就是 lastCuess 的初始值。如果 nextGUess 和 lastGuess 的差小于一个很小的数，比如 0.0001，就可以认为 nextGuess 是 n 的平方根的近似值；否则，nextGuess 就成为 lastGuess，近似过程继续执行。实现下面的方法，返回 n 的平方根。

   public static double sqrt(long n)
3. （反素数）反素数（反转拼写的素数）是指一个非回文素数，将其反转之后也是一个索数。例如：17 是一个素数，而 71 也是一个索数，所以 17 和 71 是反素数。编写程序，显示前 100 个反素数。每行显示 10 个，并且数字间用空格隔开，如下所示:

   ```
   13 17 31 37 71 73 79 97 107 113
   149 157 167 179 199 311 337 347 359 389
   ...
   ```
4. （梅森素数）如果一个素数可以写成 2^p-1 的形式，其中 p 是某个正整数，那么这个素数就称作梅森素数。编写程序，找出 p≤31 的所有梅森素数，然后显示如下的输出结果：

p	2^p -1
2	3
3	7
5	31
...	
5. （游戏：双骰儿赌博）掷双骰子游戏是赌场中非常流行的骰子游戏。编写程序，玩这个游戏的一个变种，如下所示：

   掷两个骰子。每个骰子有六个面，分别表示值 1，2，……，6。检查这两个骰子的和。如果和为 2、3 或 12（称为掷骰子（craps）），你就输了；如果和是 7 或者 11（称作自然（natural）），你就赢了；但如果和是其他数字（例如：4、5、6、8、9 或者 10），就确定了一个点。继续掷骰子，直到掷出一个 7 或者掷出和刚才相同的点数。如果掷出的是 7，你就输了。如果掷出的点数和你前一次掷出的点数相同，你就赢了。程序扮演一个独立的玩家。下面是一些运行示例。

```
You rolled 5 + 6 = 11
You win
```

```
You rolled 1 + 2 = 3
You lose
```

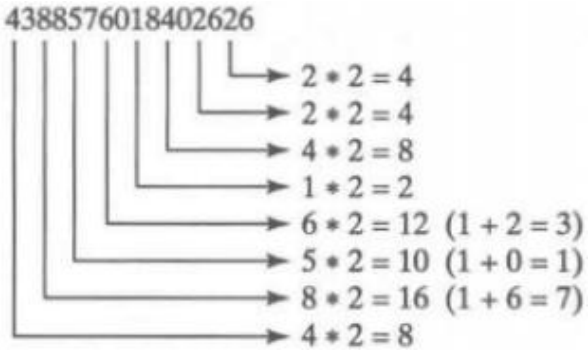
```
You rolled 4 + 4 = 8
point is 8
You rolled 6 + 2 = 8
You win
```

```
You rolled 3 + 2 = 5
point is 5
You rolled 2 + 5 = 7
You lose
```

6. （财务应用程序：信用卡号的合法性）信用卡号遵循下面的模式。一个信用卡号必须是13 到 16 位的整数。它的开头必须是：
   - 4，指 Visa 卡
   - 5，指 Master 卡
   - 37，指 American Express 卡
   - 6，指 Discover 卡

   在 1954 年，IBM 的 Hans Luhn 提出一种算法，该算法可以验证信用卡号的有效性。这个算法在确定输入的卡号是否正确，或者这张信用卡是否被扫描仪正确扫描方面是非常有用的。遵循这个合法性检测可以生成所有的信用卡号，通常称之为 Luhn 检测或者 Mod 10 检测，可以如下描述（为了方便解释，假设卡号为 4388576018402626）：

   1）从左到右对每个数字翻倍。如果对某个数字翻倍之后的结果是一个两位数，那么就将这两位加在一起得到一位数。



   2）现在将第一步得到的所有一位数相加。

$$4 + 4 + 8 + 2 + 3 + 1 + 7 + 8 = 37$$

   3）将卡号里从左到右在奇数位上的所有数字相加。

$$6 + 6 + 0 + 8 + 0 + 7 + 8 + 3 = 38$$

   4）将第二步和第三步得到的结果相加。

$$37 + 38 = 75$$

   5）如果第四步得到的结果能被 10 整除，那么卡号是合法的；否则，卡号是不合法的。例如，号码 4388576018402626 是不合法的，但是号码 4388576018410707 是合法的。编写程序，提示用户输入一个 long 型整数的信用卡号码，显示这个数字是合法的还是非法的。使用下面的方法设计程序：

```
/** Return true if the card number is valid */
public static boolean isValid(long number)

/** Get the result from Step 2 */
public static int sumOfDoubleEvenPlace(long number)

/** Return this number if it is a single digit, otherwise,
 * return the sum of the two digits */
public static int getDigit(int number)

/** Return sum of odd-place digits in number */
public static int sumOfOddPlace(long number)

/** Return true if the digit d is a prefix for number */
public static boolean prefixMatched(long number, int d)

/** Return the number of digits in d */
public static int getSize(long d)

/** Return the first k number of digits from number. If the
 * number of digits in number is less than k, return number. */
public static long getPrefix(long number, int k)
```

下面是程序的运行示例：（你也可以通过将输入作为一个字符串输入，以及对字符串进行处理来验证信用卡卡号。）

```
Enter a credit card number as a long integer:
  4388576018410707 Enter
4388576018410707 is valid
```

```
Enter a credit card number as a long integer:
  4388576018402626 Enter
4388576018402626 is invalid
```

7. （打印不同的数）编写一个程序，输入 10 个数并且显示互不相同的数（即一个数出现多次，但仅显示一次）。（提示，输入一个数，如果它是一个新数，则将它存储在数组中。如果该数已经在数组中，则忽略它。）输入之后，数组包含的都是不同的数。下面是这个程序的运行示例：

```
Enter ten numbers: 1 2 3 2 1 6 3 4 5 2 Enter
The number of distinct number is 6
The distinct numbers are: 1 2 3 6 4 5
```

8. （统计一位数的个数）编写一个程序，生成 0 和 9 之间的 100 个随机整数，然后显示每一个数出现的次数。

   提示：使用(int)(Math.random()*10)产生 0 到 9 之间的随机整数。使用一个名为 counts 的由 10 个整数构成的数组存放 0，1，…，9 的个数。

9. （随机数选择器）编写一个方法，返回 1 到 54 之间的随机数，不包括传递到参数中的 numbers。如下指定这个方法头：

```
public static int getRandom(int... numbers)
```

10. （消除重复）使用下面的方法头编写方法，消除数组中重复出现的值：

```
public static int[] eliminateDuplicates(int[] list)
```

编写一个测试程序，读取 10 个整数，调用该方法，然后显示结果。下面是程序的运行示例：

```
Enter ten numbers: 1 2 3 2 1 6 3 4 5 2 ↵Enter
The distinct numbers are: 1 2 3 6 4 5
```

11. （是否排好序了？）编写以下方法，如果参数中的 list 数组已经按照升序排好了，则返回 true。

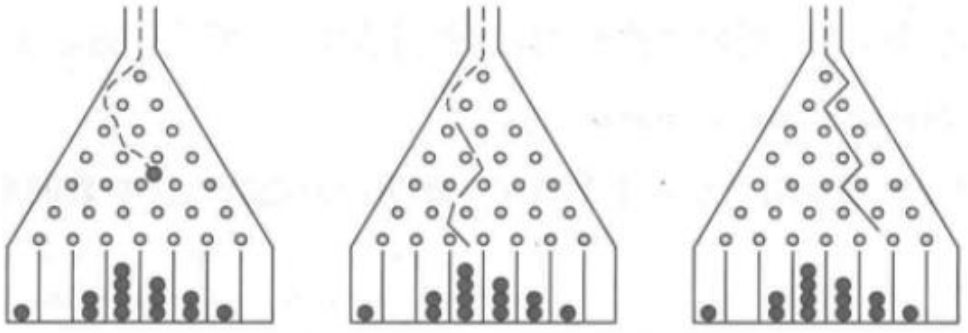
```
public static boolean isSorted(int[] list)
```

编写一个测试程序，提示用户输入一个列表，显示该列表是否已经排好序。下面是一个运行示例。注意，输入中的第一个数表示列表中的元素个数。该数不是列表的一部分。

```
Enter list: 8 10 1 5 16 61 9 11 1 ↵Enter
The list is not sorted
```

```
Enter list: 10 1 1 3 4 4 5 7 9 11 21 ↵Enter
The list is already sorted
```

12. （游戏：豆机）豆机，也称为梅花瓶或高尔顿瓶，它是一个用来做统计实验的设备，是用英国科学家瑟弗兰克斯高尔顿的名字来命名的。它是一个三角形状的均匀放置钉子（或钩子）的直立板子，如下图所示。



每个球都选取一个随机路径，然后掉入一个槽中

球都是从板子口落下的。每当球碰到钉子，它就有 50%的机会落向左边或落向右边。在板子底部的槽子中都会累积一堆球。编写程序模拟豆机。程序应该提示用户输入球的个数以及机器的槽数。打印每个球的路径模拟它的下落。例如：在图 b 中球的路径是 LLRRLLR，而在图 c 中球的路径是 RLRRLRR。使用条形图显示槽中球的最终储备量。下面是程序的一个运行示例：

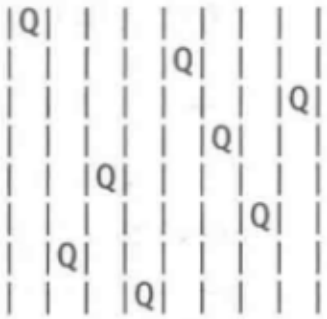
```
Enter the number of balls to drop: 5 ↵Enter
Enter the number of slots in the bean machine: 8 ↵Enter

LRLRLRR
RRLLLRR
LLRLLRR
RRLLLLL
LRLRRLR

  0
  0
000
```

提示：创建一个名为 slots 的数组。数组 slots 中的每个元素存储的是一个槽中球的个数。每个球都经过一条路径落入一个槽中。路径上 R 的个数表示球落下的槽的位置。例如：对于路径 LRLRLRR 而言，球落到 slots[4]中，而对路径 RRLLLLL 而言，球落到 slots[2]中。

13. （游戏：八皇后）经典的八皇后难题是要将八个皇后放在棋盘上，任何两个皇后都不能互相攻击（即没有两个皇后是在同一行、同一列或者同一对角上）。可能的解决方案有很多。编写程序显示一个这样的解决方案。一个示例输出如下图所示。



14. （游戏：储物柜难题）一个学校有 100 个储物柜和 100 个学生。所有的储物柜在上学第一天都是关着的。随着学生进来，第一个学生（用 S1 表示）打开每个柜子。然后，第二个学生（用 S2 表示）从第二个柜子（用 L2 表示）开始。关闭相隔为 1 的柜子。学生 S3 从第三个柜子开始，然后改变每个第三个柜子（如果它是开的就关上，如果它是关的就打开）。学生 S4 从柜子 L4 开始，然后改变每个第四个柜子的开闭状态。学生 S4 从 L4 开始，然后改变每个第五个柜子的状态，依此类推，直到学生 S100 改变 L100 为止。在所有学生都经过教学楼并且改变了柜子之后，哪些柜子是开的？编写程序找出答案。
提示：使用存放 100 个布尔型元素的数组，每个元素都表明一个柜子是开的（true）还是关的（false）。初始状态时，所有的柜子都是关的。

15. （仿真：优惠券收集人问题）优惠券收集人问题是一个经典的统计问题，它有很多实际应用。这个问题重复地从一套对象中拿出一个对象，然后找出要将所有需要拿出的对象都至少拿出来一次，需要拿多少次。从该问题衍生出的类似问题就是，从一副打乱的 52 张牌中重复选牌，找出在看到每种花色都有一张出现前，需要选多少次。假设在选下一张牌之前的那张牌是背面向上的。编写程序，模拟要得到四张不同花色的牌所需要的选取次数，然后显示选中的四张牌（有可能一张牌被选了两次）。下面是这个程序的运行示例：

```
Queen of Spades
5 of Clubs
Queen of Hearts
4 of Diamonds
Number of picks: 12
```

16. （完全相同的数组）如果两个数组 listl 和 list2 的长度相同，而且对于每个 i，list1[i]都等于 list2[i]，那么认为 listl 和 list2 是完全相同的。使用下面的方法头编写一个方法，如果 listl 和 list2 完全相同，那么这个方法返回 true：

```
public static boolean equals(int[] list1, int[] list2)
```

编写一个测试程序，提示用户输入两个整数列表，然后显示这两个列表是否完全相同。下面是运行示例。注意，输入的第一个数字表明列表中元素的个数。该数字不是列表的一部分。

```
Enter list1: 5 2 5 6 1 6 Enter
Enter list2: 5 2 5 6 1 6 Enter
Two lists are strictly identical
```

```
Enter list1: 5 2 5 6 6 1 Enter
Enter list2: 5 2 5 6 1 6 Enter
Two lists are not strictly identical
```

17. （数学方面：组合）编写一个程序，提示用户输入 10 个整数，然后显示从这 10 个数中选出两个数的所有组合。
18. （游戏：选出四张牌）编写一个程序，从一'副 52 张的牌中选出四张，然后计算它们的和。Ace、King，Queen 和 Jack 分别表示 1、13、12 和 11。程序应该显示得到的和为 24 的选牌次数。
19. （模式识别方面：四个连续相等的数）编写下面的方法，测试某个数组是否有四个连续的值相同的数字。

```
public static boolean isConsecutiveFour(int[] values)
```

编写测试程序，提示用户输入一个整数列表，如果这个列表中有四个连续的具有相同值的数，那就显示 true；否则，显示 false。程序应该首先提示用户键人输入的大小，即列表中值的个数。下面是运行示例。

```
Enter the number of values: 8 Enter
Enter the values: 3 4 5 5 5 5 4 5 Enter
The list has consecutive fours
```

```
Enter the number of values: 9 Enter
Enter the values: 3 4 5 5 6 5 5 4 5 Enter
The list has no consecutive fours
```

20. （划分列表）编写以下方法，使用第一个元素对列表进行划分，该元素称为支点。

publicstaticintpartition（int[]list）

划分后，列表中的元素被重新安排，在支点元素之前的元素都小于或者等于该元素，而之后的元素都大于该元素。方法返回支点元素位于新列表中的下标。例如，假设列表是[5，2，9，3，81，划分后.列表变为[3，2，5，9，6，8]。最多进行 list.length 次比较来

实现该方法。编写一个测试程序，提示用户输入一个列表，然后显示划分后的列表。下面是一个运行示例。注意，输入的第一个数字表示列表中元素的个数。该数字不是列表的一部分。

21. （对字符串中的字符排序）使用以下方法头编写一个方法，返回一个排好序的字符串。

```
public static String sort(String s)
```

例如.，sort（"acb"）返回 abc。编写一个测试程序。提示用户输入一个字符串，显示排好序的字符串。

22. （游戏：猜字游戏）编写一个猜字游戏。随机产生一个单词，提示用户一次猜测一个字母，如运行示例所示。单词中的每个字母显示为一个星号。当用户猜测正确后，正确的字母显示出来。当用户猜出一个单词，显示猜错的次数，并且询问用户是否继续对另外一个单词进行游戏。声明一个数组来存储单词，如下所示：

```
// Add any words you wish in this array
String[] words = {"write", "that", ...};
```

```
(Guess) Enter a letter in word ******* > p Enter
(Guess) Enter a letter in word p****** > r Enter
(Guess) Enter a letter in word pr**r** > p Enter
     p is already in the word
(Guess) Enter a letter in word pr**r** > o Enter
(Guess) Enter a letter in word pro*r** > g Enter
(Guess) Enter a letter in word progr** > n Enter
     n is not in the word
(Guess) Enter a letter in word progr** > m Enter
(Guess) Enter a letter in word progr*m > a Enter
The word is program. You missed 1 time
Do you want to guess another word? Enter y or n>
```

23. （求矩阵中各列数字的和）编写一个方法，求整数矩阵中特定列的所有元素的和，使用下面的方法头：

```
public static double sumColumn(double[][] m, int columnIndex)
```

编写一个测试程序，读取一个 3*4 的矩阵，然后显示每列元素的和。下面是一个运行示例：

```
Enter a 3-by-4 matrix row by row:
1.5 2 3 4 Enter
5.5 6 7 8 Enter
9.5 1 3 1 Enter
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9.0
Sum of the elements at column 2 is 13.0
Sum of the elements at column 3 is 13.0
```

24. （求矩阵主对角线元素的和）编写一个方法，求 n*n 的 double 类型矩阵中主对角线上所有数字的和，使用下面的方法头：

```
public static double sumMajorDiagonal(double[][] m)
```

编写一个测试程序，读取一个 4*4 的矩阵，然后显示它的主对角线上的所有元素的和。下面是一个运行示例：

```
Enter a 4 by 4 matrix row by row:
1 2 3 4.0
5 6.5 7 8
9 10 11 12
13 14 15 16
Sum of the elements in the major diagonal is 34.5
```

25. （计算每个雇员每周工作的小时数）假定所有雇员每周工作的小时数存储在一个二维数组中。每行将一个雇员 7 天的工作时间记录在 7 列中。例如：下图显示的数组存储了 8 个雇员的工作时间。

	Su	M	T	W	Th	F	Sa
Employee 0	2	4	3	4	5	8	8
Employee 1	7	3	4	3	3	4	4
Employee 2	3	3	4	3	3	2	2
Employee 3	9	3	4	7	3	4	1
Employee 4	3	5	4	3	6	3	8
Employee 5	3	4	4	6	3	4	4
Employee 6	3	7	4	8	3	8	4
Employee 7	6	3	5	9	2	7	9

编写一个程序，按照总工时降序的方式显示雇员和他们的总工时。

26. （代数方面：两个矩阵相加）编写两个矩阵相加的方法。方法头如下：

```
public static double[][] addMatrix(double[][] a, double[][] b)
```

为了能够进行相加，两个矩阵必须具有相同的维数，并且元素具有相同或兼容的数据类型。假设 C 表示最终的矩阵，每个元素 q 就是例如，对于两个 3 * 3 的矩阵 a 和 b, c 就有：

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \end{pmatrix}$$

编写一个测试程序，提示用户输入两个 3*3 的矩阵，然后显示它们的和。下面是一个运行示例：

```
Enter matrix1: 1 2 3 4 5 6 7 8 9 Enter
Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2 Enter
The matrices are added as follows
 1.0 2.0 3.0     0.0 2.0 4.0     1.0 4.0 7.0
 4.0 5.0 6.0  +  1.0 4.5 2.2  =  5.0 9.5 8.2
 7.0 8.0 9.0     1.1 4.3 5.2     8.1 12.3 14.2
```

27. （代数方面：两个矩阵相乘）编写两个矩阵相乘的方法。方法头如下：

```
public static double[][]  multiplyMatrix(double[][] a, double[][] b)
```

为了使矩阵 a 能够和矩阵 b 相乘，矩阵 a 的列数必须与矩阵 b 的行数相同，并且两个矩阵的元素要具有相同或兼容的数据类型。假设矩阵 c 是相乘的结果，而 a 的列数是 n，

那么每个元素 c_{ij} 就是$a_{i1} \times b_{1j}+a_{i2} \times b_{2j}+\cdots+a_{in} \times b_{nj}$。例如，对于两个 3*3 的矩阵 a 和 b, c 就有：

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

这里的$c_{ij}=a_{i1} \times b_{1j}+a_{i2} \times b_{2j}+a_{i3} \times b_{3j}$。

编写一个测试程序，提示用户输入两个 3*3 的矩阵，然后显示它们的乘积。下面是一个运行示例：

```
Enter matrix1: 1 2 3 4 5 6 7 8 9 Enter
Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2 Enter
The multiplication of the matrices is
 1 2 3      0 2.0 4.0      5.3 23.9 24
 4 5 6   *  1 4.5 2.2  =   11.6 56.3 58.2
 7 8 9      1.1 4.3 5.2    17.9 88.7 92.4
```

28. （所有最近的点对）修改程序清单 8-3（参见教材 P_{250}），找出所有具有相同最小距离的点对。下面是一个运行示例：

```
Enter the number of points: 8 Enter
Enter 8 points: 0 0 1 1 -1 -1  2 2 -2 -2 -3 -3 -4 -4 5 5 Enter
The closest two points are (0.0, 0.0) and (1.0, 1.0)
The closest two points are (0.0, 0.0) and (-1.0, -1.0)
The closest two points are (1.0, 1.0) and (2.0, 2.0)
The closest two points are (-1.0, -1.0) and (-2.0, -2.0)
The closest two points are (-2.0, -2.0) and (-3.0, -3.0)
The closest two points are (-3.0, -3.0) and (-4.0, -4.0)
Their distance is 1.4142135623730951
```

29. （游戏：井字游戏）在井字游戏中，两个玩家使用各自的标志（一方用 X 则另一方就用 O），轮流填写 3*3 的网格中的某个空格。当一个玩家在网格的水平方向、垂直方向或者对角线方向上出现了三个相同的 X 或三个相同的 O 时，游戏结束，该玩家获胜。平局（没有赢家）是指当网格中所有的空格都被填满时没有任何一方的玩家获胜的情况。创建一个玩井字游戏的程序。程序提示两个玩家可以选择 X 和 O 作为他们的标志。当输入一个标志时，程序在控制台上重新显示棋盘，然后确定游戏的状态（是获胜、平局还是继续）。下面是一个运行示例：

```
 |   |   |
 |   |   |
 |   |   |
Enter a row (0, 1, or 2) for player X: 1 Enter
Enter a column (0, 1, or 2) for player X: 1 Enter

 |   |   |
 |   | X |
 |   |   |
Enter a row (0, 1, or 2) for player O: 1 Enter
Enter a column (0, 1, or 2) for player O: 2 Enter

 |   |   |
 |   | X | O |
 |   |   |
Enter a row (0, 1, or 2) for player X:

 . . .

 | X |   |   |
 | O | X | O |
 |   |   | X |
X player won
```

30. （游戏：九个正面和背面）一个 3*3 的矩阵中放置了 9 个硬币，这些硬币有些面向上，有些面向下。可以使用 3*3 的矩阵中的 0（正面）或 1（反面）表示硬币的状态。下面是一些例子：

```
0 0 0    1 0 1    1 1 0    1 0 1    1 0 0
0 1 0    0 0 1    1 0 0    1 1 0    1 1 1
0 0 0    1 0 0    0 0 1    1 0 0    1 1 0
```

每个状态都可以使用一个二进制数表示。例如，前面的矩阵对应到数字：

```
000010000 101001100 110100001 101110100 100111110
```

总共会有 512 种可能性。所以，可以使用十进制数 0，1，2，3，511 来表示这个矩阵的所有状态。编写一个程序，提示用户输入一个在 0 到 511 之间的数字，然后显示用字符 H 和 T 表示的对应的矩阵。下面是一个运行示例：

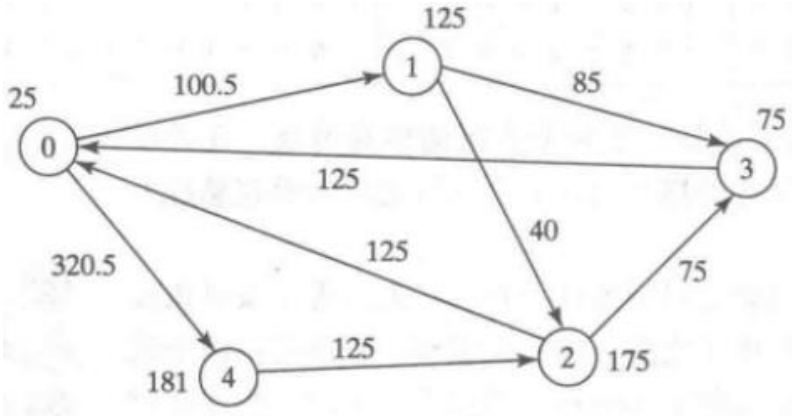
```
Enter a number between 0 and 511: 7 Enter
H H H
H H H
T T T
```

用户输入 7，它代表的是 000000111。因为 0 代表 H 而 1 代表 T，所以输出正确。

31. （探索矩阵）编写程序，提示用户输入一个方阵的长度，随机地在矩阵中填入 0 和 1，打印这个矩阵，然后找出整行、整列或者对角线都是 0 或 1 的行、列和对角线。下面是这个程序的一个运行示例：

```
Enter the size for the matrix: 4 Enter
0111
0000
0100
1111
All 0s on row 1
All 1s on row 3
No same numbers on a column
No same numbers on the major diagonal
No same numbers on the sub-diagonal
```

32. （金融风暴）银行会互相借钱。在经济艰难时期，如果一个银行倒闭，它就不能偿还贷款。一个银行的总资产是它当前的余款减去它欠其他银行的贷款。下图就是五个银行的状况图。每个银行的当前余额分别是 2500 万美元、1 亿 2500 万美元、1 亿 7500 万美元、7500 万美元和 1 亿 8100 万美元。从节点 1 到节点 2 的方向的边表示银行 1 借给银行 2 共计 4 千万美元。



银行之间互相借款

如果银行的总资产在某个限定范围以下，那么这个银行就是不安全的。它借的钱就不能返还给借贷方，而且这个借贷方也不能将这个贷款算人它的总资产。因此，如果借贷方总资产在限定范围以下，那么它也不安全。编写程序，找出所有不安全的银行。程序如下读取输入。它首先读取两个整数 n 和 limit，这里的 n 表示银行个数，而 limit 表示要保持银行安全的最小总资产。然后，程序会读取描述 n 个银行的 n 行信息，银行的 id 从 0 到 n-1。每一行的第一个数字都是银行的余额，第二个数字表明从该银行借款的银行，其余的就都是两个数字构成的数对。每对都描述一个借款方。每一对数字的第一个数就是借款方的 id，第二个数就是所借的钱数。例如，在上图中五个银行的输入如下所示（注意：limit 是 201）：

```
5 201
25 2 1 100.5 4 320.5
125 2 2 40 3 85
175 2 0 125 3 75
75 1 0 125
181 1 2 125
```

银行 3 的总资产是 75+125，这个数字是在 201 以下的。所以，银行 3 是不安全的。在银行 3 变得不安全之后，银行 1 的总资产也降为 125+40。所以，银行 1 也不安全。程序的输出应该是：

Unsafe banks are 3 1

提示：使用一个二维数组 borrowers 来表示贷款。borrowers[i][j]表明银行 i 贷款给银行 j 的贷款额。一旦银行：j 变得不安全，那么 borrowers[i][j]就应该设置为 0。

33. （模式识别：连续的四个相等的数）编写下面的方法.测试一个二维数组是否有四个连续的数字具有相同的值，这四个数可以是水平方向的、垂直方向的或者对角线方向的。

public static boolean isConsecutiveFour(int[][] values)

编写一个测试程序，提示用户输入一个二维数组的行数、列数以及数组中的值。如果这个数组有四个连续的数字具有相同的值，就显示 true；否则，显示 false。下面是结果为 true 的一些例子：

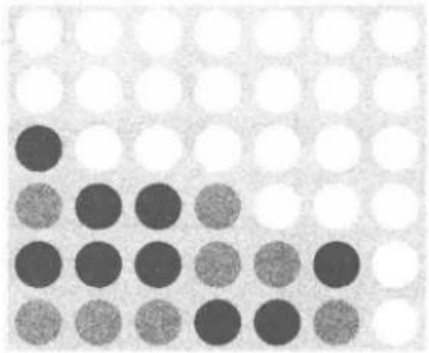
0	1	0	3	1	6	1
0	1	6	8	6	0	1
5	6	2	1	8	2	9
6	5	6	1	1	9	1
1	3	6	1	4	0	7
3	3	3	3	4	0	7

0	1	0	3	1	6	1
0	1	6	8	6	0	1
5	5	2	1	8	2	9
6	5	6	1	1	9	1
1	5	6	1	4	0	7
3	5	3	3	4	0	7

0	1	0	3	1	6	1
0	1	6	8	6	0	1
5	6	2	1	6	2	9
6	5	6	6	1	9	1
1	3	6	1	4	0	7
3	6	3	3	4	0	7

0	1	0	3	1	6	1
0	1	6	8	6	0	1
9	6	2	1	8	2	9
6	9	6	1	1	9	1
1	3	9	1	4	0	7
3	3	3	9	4	0	7

34. （游戏：四子连）四子连是一个两个人玩的棋盘游戏，在游戏中，玩家轮流将有颜色的棋子放在一个六行七列的垂直悬挂的网格中，如下图所示。



这个游戏的目的是在对手实现一行、一列或者一条对角线上有四个相同颜色的棋子之前，你能先做到。程序提示两个玩家交替地下红子 Red 或黄子 Yellow。当放下一子时，程序在控制台重新显示这个棋盘，然后确定游戏的状态（赢、平局还是继续）。下面是一个运行示例：

```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
---------------
Drop a red disk at column (0-6): 0 Enter

| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
|R| | | | | | |
---------------
Drop a yellow disk at column (0-6): 3 Enter
| | | | | | | |
| | | || | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
|R| | |Y| | | |
```

```
. . .
. . .
. . .

Drop a yellow disk at column (0-6): 6 Enter
| | | | | | | |
| | | | | | | |
| | | |R| | | |
| | | |Y|R|Y| |
| | |R|Y|Y|Y|Y|
|R|Y|R|Y|R|R|R|
---------------
The yellow player won
```

35. （中心城市）给定一组城市，中心城市是和所有其他城市之间具有最短距离的城市。编写一个程序，提示用户输入城市的数目以及城市的位置（坐标），找到中心城市以及和所有其他城市的总距离。

```
Enter the number of cities: 5 Enter
Enter the coordinates of the cities:
  2.5 5 5.1 3 1 9 5.4 54 5.5 2.1 Enter
The central city is at (2.5, 5.0)
The total distance to all other cities is 60.81
```

36. （游戏：找到翻转的单元格）假设给定一个填满 0 和 1 的 6*6 矩阵。所有的行和列都有偶数个 1。让用户翻转一个单元（即从 1 翻成 0 或者从 0 翻成 1），编写一个程序找到哪个单元格被翻转了。程序应该提示用户输入一个 6*6 的填满 0 和 1 的矩阵，并且找到第一个 r 行以及第一个 c 列具有偶数个 1 的特征是不符合的（即 1 的数目不是偶数），则该翻转的单元格位于（r，c）。下面是一个运行示例：

```
Enter a 6-by-6 matrix row by row:
1 1 1 0 1 1 Enter
1 1 1 1 0 0 Enter
0 1 0 1 1 1 Enter
1 1 1 1 1 1 Enter
0 1 1 1 1 0 Enter
1 0 0 0 0 1 Enter
The flipped cell is at (0, 1)
```

37. （马尔科夫矩阵）一个 n*n 的矩阵被称为一个正马尔科夫矩阵，当且仅当每个元素都是正数，并且每列的元素的和为 1。编写下面的方法来检测一个矩阵是否是一个马尔科夫矩阵。

    public static boolean isMarkovMatrix(double[][] m)

    编写一个测试程序，提示用户输入一个 3*3 的 double 值的矩阵，测试它是否是一个马尔科夫矩阵。下面是运行示例：

```
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375 Enter
0.55 0.005 0.225 Enter
0.30 0.12 0.4 Enter
It is a Markov matrix
```

```
Enter a 3-by-3 matrix row by row:
0.95 -0.875 0.375 Enter
0.65 0.005 0.225 Enter
0.30 0.22 -0.4 Enter
It is not a Markov matrix
```

38. （列排序）用下面的方法实现一个二维数组中的列排序。返回一个新的数组，并且原数组保持不变。

    public static double[][] sortColumns(double[][] m)

    编写一个测试程序，提示用户输入一个 3*3 的 double 类型值的矩阵，显示一个新的每列排好序的矩阵。下面是一个运行示例。

```
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375 Enter
0.55 0.005 0.225 Enter
0.30 0.12 0.4 Enter

The column-sorted array is
0.15 0.0050 0.225
0.3  0.12   0.375
0.55 0.875  0.4
```

39. （严格相同的数组）如果两个二维数组 ml 和 m2 相应的元素是相等的话，则认为它们是严格相同的。编写一个方法，如果 ml 和 m2 是严格相同的话，返回 true。使用下面的方法头：

    public static boolean equals(int[][]ml，int[][]m2)

    编写一个测试程序，提示用户输入两个 3*3 的整数数组，显示两个矩阵是否是严格相同的。下面是运行示例。

```
Enter list1: 51 22 25 6 1 4 24 54 6 Enter
Enter list2: 51 22 25 6 1 4 24 54 6 Enter
The two arrays are strictly identical
```

```
Enter list1: 51 25 22 6 1 4 24 54 6 Enter
Enter list2: 51 22 25 6 1 4 24 54 6 Enter
The two arrays are not strictly identical
```

40. （最大块）给定一个元素为 0 或者 1 的方阵，编写一个程序，找到一个元素都为 1 的最大的子方阵。程序提示用户输入矩阵的行数。然后显示最大的子方阵的第一个元素，以及该子方阵中的行数。下面是一个运行示例。

```
Enter the number of rows in the matrix: 5 Enter
Enter the matrix row by row:
1 0 1 0 1 Enter
1 1 1 0 1 Enter
1 0 1 1 1 Enter
1 0 1 1 1 Enter
1 0 1 1 1 Enter

The maximum square submatrix is at (2, 2) with size 3
```

程序需要实现和使用下面的方法来找到最大的子方阵：

public static int[] flndLargestBlock(int[][] m)

返回值是一个包含三个值的数组。前面两个值是子方阵中的行和列的下标，第 3 个值是子方阵中的行数。

41. （拉丁正方形）拉丁正方形是一个 n*n 的数组，由 n 个不同的拉丁字母填充，每个拉丁字母恰好只在每行和每列中出现一次。编写一个程序，提示用户输入数字 n 以及字符数组，如示例输出所示，检测该输出数组是否是一个拉丁正方形。字符是从 A 开始的前面 n 个字符。

```
Enter number n: 4 Enter
Enter 4 rows of letters separated by spaces:
A B C D Enter
B A D C Enter
C D B A Enter
D C A B Enter
The input array is a Latin square
```

```
Enter number n: 3 Enter
Enter 3 rows of letters separated by spaces:
A F D Enter
Wrong input: the letters must be from A to C
```

42. （猜测首府）编写一个程序，重复提示用户输入一个州的首府。当接收到用户输入后，程序报告答案是否正确。假设 50 个州以及它们的首府保存在一个二维数组中，如下图所示。程序提示用户回答所有州的首府，并且显示所有正确问答的数 B（忽略英文字母的大小写）。

Alabama	Montgomery
Alaska	Juneau
Arizona	Phoenix
...	...
...	...

一个保存了州以及它们的首府的二维数组

下面是一个运行示例。

```
What is the capital of Alabama? Montogomery ↵Enter
The correct answer should be Montgomery
What is the capital of Alaska? Juneau ↵Enter
Your answer is correct
What is the capital of Arizona? ...
...
The correct count is 35
```

实验 4 对象和类（1）

1. （矩形类 Rectangle）遵照教材 9.2 节（参见教材 P_{270}）中 Circle 类的例子，设计一个名为 Rectangle 的类表示矩形。这个类包括：
   - 两个名为 width 和 height 的 double 型数据域，它们分别表示矩形的宽和高。width 和 height 的默认值都为 1。
   - 创建默认矩形的无参构造方法。
   - —个创建 width 和 height 为指定值的矩形的构造方法。
   - 一个名为 getArea（）的方法返回这个矩形的面积。
   - 一个名为 getPerimeter（）的方法返回周长。

   实现这个类并编写一个测试程序，创建两个 Rectangle 对象—一个矩形的宽为 4 而高为 40，另一个矩形的宽为 3.5 而高为 35.9。按照这个顺序显示每个矩形的宽、高、面积和周长。
2. （股票类 Stock）遵照教材 9.2 节（参见教材 P_{270}）中 Circle 类的例子，设计一个名为 Stock 的类。这个类包括：
   - —个名为 symbol 的字符串数据域表示股票代码。
   - 一个名为 name 的字符串数据域表示股票名字。
   - —个名为 previousClosingPrice 的 double 型数据域，它存储的是前一日的股票值。
   - —个名为 currentPrice 的 double 型数据域，它存储的是当时的股票值。
   - 创建一支有特定代码和名字的股票的构造方法。
   - 一个名为 getChangePercent（）的方法，返回从 previousClosingPrice 变化到 currentPrice 的百分比。

   实现这个类，并编写一个测试程序，创建一个 Stock 对象，它的股票代码是 ORCL，股票名字为 OracleCorporation，前一日收盘价是 34.5。设置新的当前值为 34.35，然后显示市值变化的百分比。
3. （使用日期类 Date）编写程序创建一个 Date 对象，设置它的流逝时间分别为 10000、100000、1000000、10000000、100000000、1000000000、10000000000、100000000000，然后使用 toString（）方法分别显示上述日期。
4. （使用随机类 Random）编写一个程序，创建种子是 1000 的 Random 对象，然后使用 nextlnt（l00）方法显示 0 到 100 之间前 50 个随机整数。
5. （使用公历类 GregorianCalendar）JavaAPI 有一个在包 java.util 中的类 GregorianCalendar，可以使用它获得某个日期的年、月、日。它的无参构造方法构建一个当前日期的实例，get （ GregorianCalendar.YEAR ）、 get （ GregorianCalendar.MONTH ） 和 get （GregorianCalendar.DAY_OF_M0NTH）方法返回年、月和日。编写一个程序完成两个任务：
   - 显示当前的年、月和日。
   - GregorianCalendar 类有方法 setTimelnMillis(long)，可以用它来设置从 1970 年 1 月 1 日算起的一个特定时间。将这个值设置为 1234567898765L，然后显示这个年、月和日。
6. （秒表）设计一个名为 Stopwatch 的类。该类包含：
   - 具有访问器方法的私有数据域 startTime 和 endTime。
   - 一个无参构造方法，使用当前时间来初始化 startTime。
   - 一个名为 start（）的方法，将 startTime 重设为当前时间。

- 一个名为 stop（）的方法，将 endTime 设置为当前时间。
- —个名为 getElapsedTime（）的方法，以毫秒为单位返回秒表记录的流逝时间。

编写一个测试程序，用于测量使用选择排序对 100000 个数字进行排序的执行时间。

7. （账户类 Account）设计一个名为 Account 的类，它包括：
   - —个名为 id 的 int 类型私有数据域（默认值为 0）。
   - —个名为 balance 的 double 类型私有数据域（默认值为 0）。
   - —个名为 annualInterestRate 的 double 类型私有数据域存储当前利率（默认值为 0）。假设所有的账户都有相同的利率。
   - —个名为 dateCreated 的 Date 类型的私有数据域，存储账户的开户日期。
   - —个用于创建默认账户的无参构造方法。
   - 一个用于创建带特定 id 和初始余额的账户的构造方法。
   - id、balance 和 annualInterstRate 的访问器和修改器。
   - dateCreated 的访问器。
   - 一个名为 getMonthlyInterestRate（）的方法，返回月利率。
   - —个名为 withDraw 的方法，从账户提取特定数额。
   - —个名为 deposit 的方法向账户存储特定数额。

   实现这个类。提示：方法 getMonthlyInterest()用于返回月利息，而不是利率。月利息是 balance*monthly-InterestRate 。 monthlyInterestRate 是 annualInterestRate/12 。 注 意 ， annualInterestRate 是一个百分数，比如 4.5%。你需要将其除以 100。

   编写一个测试程序，创建一个账户 ID 为 1122、余额为 20000 美元、年利率为 4.5%的 Account 对象。使用 withdraw 方法取款 2500 美元，使用 deposit 方法存款 3000 美元，然后打印余额、月利息以及这个账户的开户日期。
8. （风扇类 Fan）设计一个名为 Fan 的类来表示一个风扇。这个类包括：
   - 三个名为 SLOW、MEDIUM 和 FAST 而值为 1、2 和 3 的常量，表示风扇的速度。
   - —个名为 speed 的 int 类型私有数据域，表示风扇的速度（默认值为 SLOW）。
   - —个名为 on 的 boolean 类型私有数据域，表示风扇是否打开（默认值为 false）。
   - 一个名为 radius 的 double 类型私有数据域，表示风扇的半径（默认值为 5）。
   - 一个名为 color 的 string 类型数据域，表示风扇的颜色（默认值为 blue）。
   - 这四个数据域的访问器和修改器。
   - 一个创建默认风扇的无参构造方法。
   - 一个名为 toString（）的方法返回描述风扇的字符串。如果风扇是打开的，那么该方法在一个组合的字符串中返回风扇的速度、颜色和半径。如果风扇没有打开，该方法就会返回一个由“fan is off”和风扇颜色及半径组合成的字符串。

   编写一个测试程序，创建两个 Fan 对象。将第一个对象设置为最大速度、半径为 10、颜色为 yellow、状态为打开。将第二个对象设置为中等速度、半径为 5、颜色为 blue、状态为关闭。通过调用它们的 toString 方法显示这些对象。
9. （几何：正 n 边形）在一个正 n 边形中，所有边的长度都相同，且所有角的度数都相同（即这个多边形是等边等角的）。设计一个名为 RegularPolygon 的类，该类包括：
   - 一个名为 n 的 int 型私有数据域定义多边形的边数，默认值为 3。
   - 一个名为 side 的 double 型私有数据域存储边的长度，默认值为 1。
   - 一个名为 x 的 double 型私有数据域定义多边形中点的 x 坐标，默认值为 0。
   - 一个名为 y 的 double 型私有数据域定义多边形中点的 y 坐标，默认值为 0。
   - 一个创建带默认值的正多边形的无参构造方法。
   - 一个能创建带指定边数和边长度、中心在（0，0）的正多边形的构造方法。

- 一个能创建带指定边数和边长度、中心在（x，y）的正多边形的构造方法。
- 所有数据域的访问器和修改器。
- —个返回多边形周长的方法 getPerimeter（）。
- 一个返回多边形面积的方法 getArea（）。计算正多边形面积的公式是：

$$面积=\frac{n\times s^2}{4\times\tan\left(\frac{\pi}{n}\right)}$$

编写一个测试程序，分别使用无参构造方法、RegularPo1ygon（6，4）和 RegularPolygon（10，4，5_6，7_8）创建三个 RegularPolygon 对象。显示每个对象的周长和面积。

10. （代数：二次方程式）为二次方程式 $ax^2+bx+c=0$ 设计一个名为 QuadraticEquation 的类。这个类包括：
    - 代表三个系数的私有数据域 a、b 和 c。
    - 一个参数为 a、b 和 c 的构造方法。
    - a、b、c 的三个 get 方法。
    - —个名为 getDiscriminant（）的方法返回判别式，b^2-4ac
    - 名为 getRootl（）和 getRoot2（）的方法返回等式的两个根：

$$r_1=\frac{-b+\sqrt{b^2-4ac}}{2a}\quad 和\quad r_2=\frac{-b-\sqrt{b^2-4ac}}{2a}$$

    这些方法只有在判别式为非负数时才有用。如果判别式为负，这些方法返回 0。
    编写一个测试程序，提示用户输入 a、b 和 c 的值，然后显示判别式的结果。如果判别式为正数，显示两个根；如果判别式为 0，显示一个根；否则，显示“The equation has no roots.”（这个方程无根）。
11. （代数：2*2 的线性方程）为一个 2*2 的线性方程设计一个名为 LinearEquation 的类：

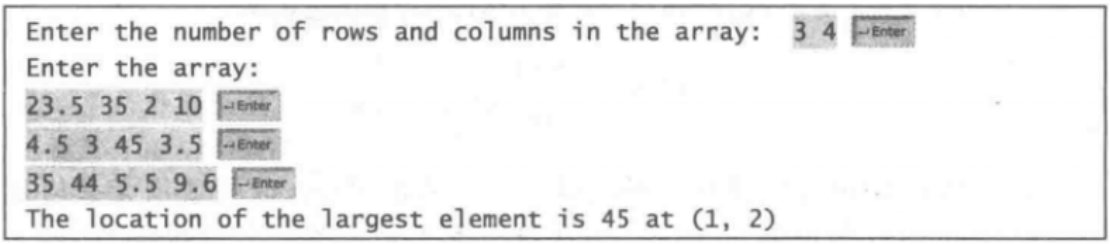
$$\begin{aligned}ax+by&=e\\cx+dy&=f\end{aligned}\qquad x=\frac{ed-bf}{ad-bc}\qquad y=\frac{af-ec}{ad-bc}$$

    这个类包括：
    - 私有数据域 a、b、c、d、e 和 f。
    - 一个参数为 a、b、c、d、e、f 的构造方法。
    - a、b、c、d、e、f 的六个 get 方法。
    - 一个名为 isSolvable（）的方法，如果 ad-bc 不为 0 则返回 true。
    - 方法 getX（）和 getY（）返回这个方程的解。

    编写一个测试程序，提示用户输入 a、b、c、d、e、f 的值，然后显示它的结果。如果 ad-bc 为 0，就报告“The equation has no solution.”。
12. （几何：交点）假设两条线段相交。第一条线段的两个端点是（xl，yl）和（x2，y2），第二条线段的两个端点是（x3，y3）和（x4，y4）。编写一个程序，提示用户输入这四个端点，然后显示它们的交点。可以通过对一个线性方程求解来得到。
13. （位置类 Location）设计一个名为 Location 的类，定位二维数组中的最大值及其位置。这个类包括公共的数据域 row、column 和 maxValue，二维数组中的最大值及其下标用 int 型的 row 和 column 以及 double 型的 maxValue 存储。编写下面的方法，返回一个二维数组中最大值的位置。

    public static Location locateLargest（double[][] a）

返回值是一个 Location 的实例。编写一个测试程序，提示用户输入一个二维数组.然后显示这个数组中最大元素的位置。下面是一个运行示例：



实验 5 对象和类（2）

1. (时间类 Time)设计一个名为 Time 的类。这个类包含：

- 表示时间的数据域 hour、minute 和 second。
- 一个以当前时间创建 Time 对象的无参构造方法（数据域的值表示当前时间）。
- 一个构造 Time 对象的构造方法，这个对象有一个特定的时间值，这个值是以毫秒表示的、从 1970 年 1 月丨日午夜开始到现在流逝的时间段（数据域的值表示这个时间）。
- 一个构造带特定的小时、分钟和秒的 Time 对象的构造方法。
- 三个数据域 hour、minute 和 second 各自的 get 方法。
- 一个名为 setTime(long elapseTime)的方法，使用流逝的时间给对象设置一个新时间。例如，如果流逝的时间为 555550000 毫秒，则转换为 10 小时、10 分钟、10 秒。

实现这个类。编写一个测试程序，创建两个 Time 对象（使用 newTime()和 new Time(555550000)), 然后显示它们的小时、分钟和秒。

提示：前两个构造方法可以从流逝的时间中提取出小时、分钟和秒。对于无参构造方法，当前时间可以使用 System.currentTimeMills()获取当前时间。

2. (BMI 类）将下面的新构造方法加人到 BMI 类中：

```
/** Construct a BMI with the specified name , age, weight , feet, and inches */
    public BMI(String name , int age , double weight, double feet,double inches)
```

3. （Mylnteger 类）设计一个名为 Mylnteger 的类。这个类包括：

- 一个名为 value 的 int 型数据域，存储这个对象表示的 int 值。
- 一个为指定的 int 值创建 MyInteger 对象的构造方法。
- 一个返回 int 值的 get 方法。
- 如果值分别为偶数、奇数或素数，那么 isEven()、isOdd()和 isPrime()方法都会返回 true。
- 如果指定值分别为偶数、奇数或素数，那么相应的静态方法 isEven(int)、isOdd(int)和 isPrime(int)会返回 true。
- 如果指定值分别为偶数、奇数或素数，那么相应的静态方法 isEven(Mylnteger)、isOdd(Mylnteger)和 isPrime(Mylnteger)会返回 true。
- 如果该对象的值与指定的值相等，那么 equals(int)和 equals(Mylnteger)方法返回 true。
- 静态方法 parselnt(char[])将数字字符构成的数组转换为一个 int 值。
- 静态方法 parselnt(String)将一个字符串转换为一个 int 值。

实现这个类并编写客户程序测试这个类中的所有方法。

4. (MyPoint 类）设计一个名为 MyPoint 的类，表示一个带 x 坐标和 y 坐标的点。该类包括：

- 两个带 get 方法的数据域 x 和 y 分别表示它们的坐标。
- 一个创建点(0,0)的无参构造方法。
- 一个创建特定坐标点的构造方法。
- — 个名为 distance 的方法，返回从该点到 MyPoint 类型的指定点之间的距离。
- 一个名为 distance 的方法，返回从该点到指定 x 和 y 坐标的指定点之间的距离。

实现这个类并编写一个测试程序，创建两个点（0,0)和（10,30.5)，并显示它们之间的距离。

5. （显示素数因子）编写一个程序，提示用户输入一个正整数，然后以降序显示它的所有最小因子。例如：如果整数为 120, 那么显示的最小因子为 5、3、2、2、2。使用 StackOfIntegers 类存储因子（例如：2、2、2、3、5), 获取之后按倒序显示这些因子。

6. （显示素教)编写一个程序，然后按降序显示小于 120 的所有素数。使用 StackOflntegers

类存储这些素数（例如：2、3、5、...)，获取之后按倒序显示它们。

7. （游戏：ATM 机）使用教材编程练习题 9.7（参见教材 P_{306}）中创建的 Account 类来模拟一台 ATM 机。创建一个有 10 个账户的数组，其 id 为 0,1,…，9, 并初始化收支为 100 美元。系统提示用户输人一个 id。如果输人的 id 不正确，就要求用户输人正确的 id。一旦接受一个 id, 就显示如运行示例所示的主菜单。可以选择 1 来查看当前的收支，选择 2 表示取钱，选择 3 表示存钱，选择 4 表示退出主菜单。一旦退出，系统就会提示再次输入 id。所以，系统一旦启动就不会停止。

```
Enter an id: 4 Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 Enter
The balance is 100.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 2 Enter
Enter an amount to withdraw: 3 Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 Enter
The balance is 97.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 3 Enter
Enter an amount to deposit: 10 Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 Enter
The balance is 107.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 4 Enter

Enter an id:
```

8. （财务：税款类 Tax）教材编程练习题 8.12（参见教材 P_{261}）使用数组编写一个计算税

款的程序。设计一个名为 Tax 的类，该类包含下面的实例数据域。

- int filingStatus（四种纳税人状态之一）：0—单身纳税人、1—已婚共缴纳税人或合法寡妇、2—已婚单缴纳税人、3—家庭纳税人。使用公共静态常量 SINGLE_FILER(0)、MARRIED_JOINTLY_OR_QUALIFYING_WIDOW(ER) (1)、MARRIED_SEPARATELY(2) 和 HEAD_OF_HOUSEHOLD(3) 表示这些状态。
- int[][] brackets：存储每种纳税人的纳税等级。
- double[] rates：存储每种纳税等级的税率。
- double taxableIncome：存储可征税收入。

给每个数据域提供 get 和 set 方法，并提供返回税款的 getTax()方法。该类还提供一个无参构造方法和构造方法 Tax(filingStatus，brackets，rates，taxableIncome)。

实现这个类并编写一个测试程序，使用 Tax 类对所给四种纳税人打印 2001 年和 2009 年的税款表，可征税收入范围在 50 000 美元和 60 000 美元之间，间隔区间为 1000 美元。2009 年的税率参见教材表 3-2（参见教材 P_{77}）, 2001 年的税率参见下表。

2001 年美国联邦个人所得税税率表

税率	单身纳税人	已婚共缴纳税人或符合条件的丧偶人士	已婚单缴纳税人	家庭纳税人
15%	$27 050 以下	$45 200 以下	$22 600 以下	$36 250 以下
27.5%	$27 051 ～ $65 550	$45 201 ～ $109 250	$22 601 ～ $54 625	$36 251-$93 650
30.5%	$65 551 ～ $136 750	$109 251 ～ 166 500	$54 626 ～ $83 250	$93 651-$151 650
35.5%	$136 751 ～ $29 7350	$166 501 ～ $297 350	$83 251 ～ $148 675	$151 651-$297 350
39.1%	$297 351 及以上	$297 351 及以上	$148 676 及以上	$297 351 及以上

9. （课程类 Course）如下改写 Course 类：

- 程序清单 10-6（参见教材 P_{319}）中数组的大小是固定的。对它进行改进，通过创建一个新的更大的数组并复制当前数组的内容来实现数组大小的自动增长。
- 实现 dropStudent 方法。
- 添加一个名为 clear()的新方法，然后删掉选某门课程的所有学生。

编写一个测试程序，创建一门课程，添加三个学生，删除一个学生，然后显示这门课程的学生。

10. （Queue 类）教材 10.6 节（参见教材 P_{320}）给出了一个用于 Stack 的类。设计一个名为 Queue 的类用于存储整数。像栈一样，队列具有元素。在栈中，元素以“后进先出”的方式获得。在队列中，元素以“先进先出”的方式获取。该类包含：

- 一个名为 element 的 int[]类型的数据域，保存队列中的 int 值。
- 一个名为 size 的数据域，保存队列中的元素个数。
- 一个构造方法，使用默认的容量 8 来创建一个 Queue 对象。
- 方法 enqueue(int v)，用于将 v 加人到队列中。
- 方法 dequeue()，用于从队列中移除元素并返回该元素。
- 方法 empty()，如果队列是空的话，该方法返回 true。
- 方法 getSize()，返回队列的大小。

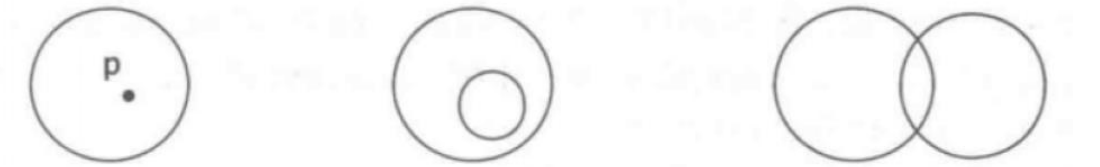
实现这个类，使之初始数组的大小为 8。一旦元素个数超过了大小，数组大小将会翻倍。如果一个元素从数组的开始部分移除，你需要将数组中的所有元素往左边改变一个位置。编写一个测试程序，增加从 1 到 20 的 21 个成员，然后将这些数字移除并显示它们。

11. （几何：Circle2D 类）定义 Circle2D 类，包括：

- 两个带有 get 方法的名为 x 和 y 的 double 型数据域，表明圆的中心点。
- 一个带 get 方法的数据域 radius。
- 一个无参构造方法，该方法创建一个(x,y)值为(0,0)且 radius 为 1 的默认圆。

- 一个构造方法，创建带指定的 x、y 和 radius 的圆。
- 一个返回圆面积的方法 getArea()。
- 一个返回圆周长的方法 getPerimeter()。
- 如果给定的点(x,y)在圆内，那么方法 contains(double x, double y)返回 true, 如下图 a 所示。
- 如果给定的圆在这个圆内，那么方法 contains(Circle2D ci rcle)返回 true, 如下图 b 所示。
- 如果给定的圆和这个圆重叠，那么方法 overlaps(Circle2D circle)返回 true, 如下图 c 所示。

实现这个类并编写测试程序，创建一个 Circle2D 对象 cl(new Circle2D(2,2,5.5)), 显示它的面积和周长，还要显示 cl.contains(3,3)、cl.contains(new Circle2D (4,5,10.5))和 cl.overlaps(new Circ1e2D(3,5,2.3))的结果。



a）点在圆内　　b）一个圆在另一个圆内　　c）一个圆和另一个圆重叠

12. （几何：Triangle2D 类）定义 Triang1e2D 类，包含：

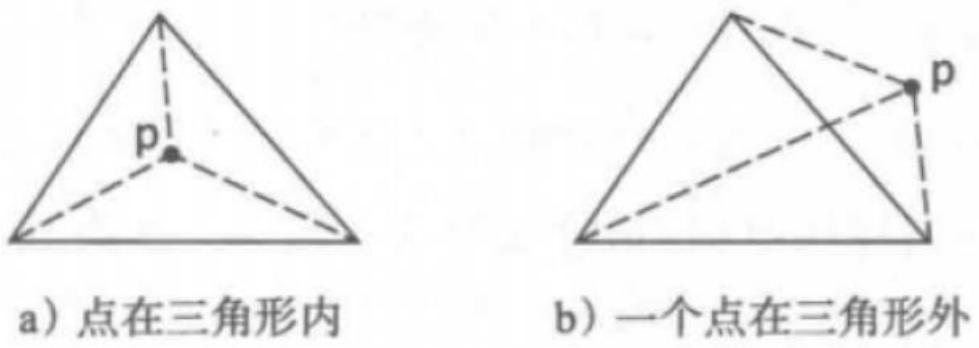
- 三个名为 pl、p2 和 p3 的 MyPoint 类型数据域，这三个数据域都带有 get 和 set 方法。MyPoint 在教材编程练习题 10.4 （参见教材 P_{340}）中定义。
- 一个无参构造方法，该方法创建三个坐标为（0,0）、（1,1）和（2,5）的点组成的默认三角形。
- 一个创建带指定点的三角形的构造方法。
- 一个返回三角形面积的方法 getArea()。
- 一个返回三角形周长的方法 getPerimeter()。
- 如果给定的点 p 在这个三角形内，那么方法 contains(MyPoint p)返回 true, 如下图 a 所示。
- 如果给定的三角形在这个三角形内，那么方法 contains(Triangle2D t)返回 true, 如下图 b 所示。
- 如果给定的三角形和这个三角形重叠，那么方法 overlaps(Triangle2D t)返回 true, 如下图 c 所示。

a）点在三角形内　　b）一个三角形在另一个三角形内　　c）一个三角形和另一个三角形重叠

实现这个类并编写测试程序，使用构造方法 new Triangle2D(new MyPoint(2.5,2), new MyPoint(4.2,3), new MyPoint(5,3.5))创建一个 Triangle2D 对象 t1, 显示它的面积和周长，并显示 tl.contains(3,3)、tl.contains(new Triangle2D(new MyPoint(2.9,2),new MyPoint(4,1), MyPoint(1,3.4)))和 tl.over1aps(new Triangle2D(new MyPoint(2,5.5)，new MyPoint(4,-3),MyPoint (2,6.5)))的结果。

提示：关于计算三角形面积的公式请参见教材编程练习题 2.19（参见教材 P_{62}）。为了检

测一个点是否在三角形中，画条虚线，如下图所示。如果点在三角形中，每条虚线应该和边相交一次。如果虚线和边相交两次，那么这个点肯定在这个三角形外。找到两条线交点的算法，参见教材编程练习题 3.25（参见教材 P_{96}）。



13. (几何：MyRectangle2D 类）定义 MyRectangle2D 类，包含：

- 两个名为 x 和 y 的 double 型数据域表明矩形的中心点，这两个数据域都带有 get 和 set 方法（假设这个矩形的边与 x 轴和 y 轴平行）。
- 带 get 和 set 方法的数据域 width 和 height。
- 一个无参构造方法，该方法创建一个（x,y）值为（0,0）且 width 和 height 为 1 的默认矩形。
- 一个构造方法，创建带指定的 x、y、width 和 height 的矩形。
- 方法 getArea（）返回矩形的面积。
- 方法 getPerimeter（）返回矩形的周长。
- 如果给定的点（x,y）在矩形内，那么方法 contains(double x, double y)返回 true，如下图 a 所示。
- 如果给定的矩形在这个矩形内，那么方法 contains(MyRectangle2D r)返回 true, 如下图 b 所示。
- 如果给定的矩形和这个矩形重叠，那么方法 overlaps(MyRectangle2D r)返回 true, 如下图 c 所示。

实现这个类。编写测试程序，创建一个 MyRectangle2D 对象 rl(new MyRectangle2D（2, 2,5.5,4.9)), 显示它的面积和周长，然后显示 rl.contains(3,3)、rl.contalns(new MyRectangle2D (4 ,5 ,10.S ,3.2)) 和 rl.overlaps (new MyRectangle2D(3,5, 2.3,5.4))的结果。

14. （MyDate 类）设计一个名为 MyDate 的类。该类包含：

- 表示日期的数据域 year、month 和 day。月份是从 0 开始的，即 0 表示一月份。
- 一个无参构造方法，该方法创建当前日期的 MyDate 对象。
- 一个构造方法，创建以从 1970 年 1 月 1 日午夜开始流逝的毫秒数为时间的 MyDate 对象。
- 一个构造方法，创建一个带指定年、月、日的 MyDate 对象。
- 三个数据域 year、month 和 day 的 get 方法。
- 一个名为 setDate（long elapsedTime)使用流逝的时间为对象设置新数据的方法。

实现这个类。编写测试程序，创建一个测试程序，创建两个 Date 对象（使用 new Date()和 new Date(34355555133101L), 然后显示它们的小时、分钟和秒。

提示：前两个构造方法将从逝去的时间中提取出年、月、日。例如：如果逝去的时间是 56155S550000 毫秒，那么年就是 1987, 月就是 9, 而天是 18。可以使用编程练习题 9.5（参见教材 P_{306}） 中讨论的 GregorianCalendar 类来简化编程。

15. （几何：边界矩形）边界矩形是指包围一个二维平面上一系列点的矩形，如图上图 d 所示。编写一个方法，为二维平面上一系列点返回一个边界矩形，如下所示：

    public static MyRectangle2D getRectangle(double[][] points)

    Rectangle2D 类在编程练习题 10.13 （参见教材 P_{343}）中定义。编写一个测试程序，提示用户输人 5 个点，然后显示边界矩形的中心、宽度以及高度。下面是一个运行示例：

```
Enter five points: 1.0 2.5 3 4 5 6 7 8 9 10 Enter
The bounding rectangle's center (5.0, 6.25), width 8.0, height 7.5
```

16. （被 2 或 3 整除）找出能被 2 或 3 整除的前 10 个数字，这些数字有 50 个十进制位数。
17. （平方数）找出大于 Long.MAX_VALUE 的前 10 个平方数。平方数是指形式为 n^2 的数。例如，4、9 以及 16 都是平方数。找到一种方法，使你的程序能快速运行。
18. （大素数）编写程序找出五个大于 Long.MAX_VALUE 的素数。
19. （Mersenne 素数）如果一个素数可以写成 2^p-1 的形式，那么该素数就称为 Mersenne 素数，其中的 p 是一个正整数。编写程序找出 p≤100 的所有 Mersenne 素数，然后显示如下所示的输出。（必须使用 BigInteger 来存储数字，因为它太大了，不能用 long 来存储。程序可能需要运行几个小时。）

p	2^p - 1
2	3
3	7
5	31
...	

20. （近似 e) 编程练习题 5.26（参见教材 P_{167}）使用下面数列近似计算 e

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots + \frac{1}{i!}$$

为了得到更好的精确度，在计算中使用 25 位精度的 BigDecimal。编写程序显示当 i=100, 200, 1000 时 e 的值。

21. （被 5 或 6 整除）找出能被 5 或 6 整除的大于 Long.MAX_VALUE 的前 10 个数字。
22. （实现 String 类）Java 库中提供了 String 类，给出你自己对下面方法的实现（将新类命名为 MyStringl)：

    public MyStringl(char[] chars);
    public char charAt(int index);
    public int length();
    public MyStringl substring(int begin, int end);
    public MyStringl toLowerCase();
    public boolean equals(MyStringl s);
    public static MyStringl valueOf(int i);

23. （实现 String 类）在 Java 库中提供了 String 类，给出你自己对下面方法的实现（将新类命名为 MyString2):

public MyString2(String s);
public int compare(String s);
public MyString2 substring(int begin);
public MyString2 toUpperCaseO;
public char[] toChars();
public static MyString2 valueOf(boolean b);

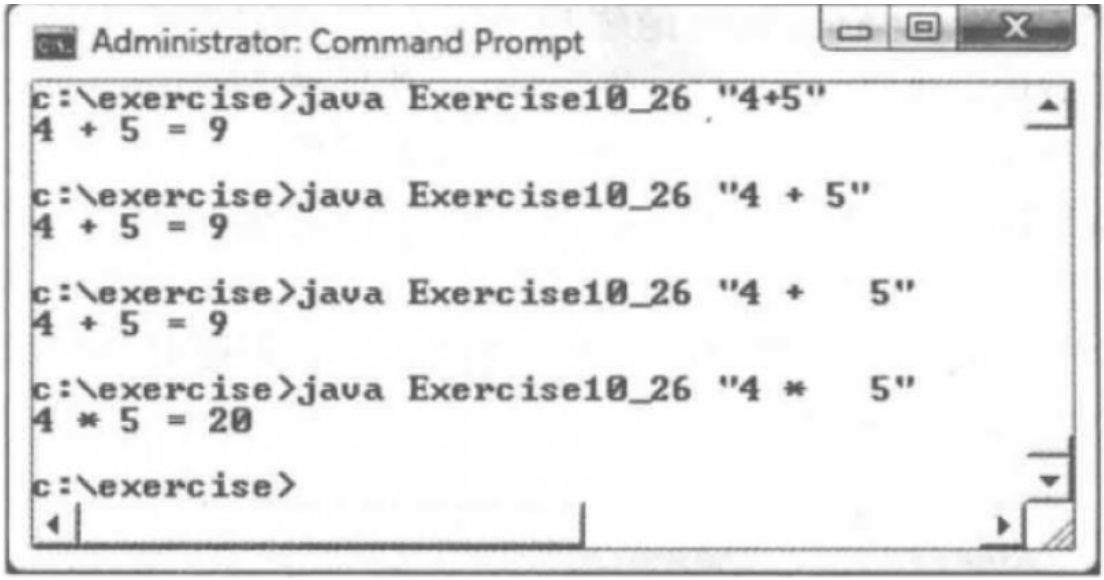
24. （实现 Character 类）在 Java 库中提供了 Character 类，给出你自己对这个类的实现（将新类命名为 MyCharacter)。

25. （新的字符串 split 方法）String 类中的 split 方法会返回一个字符串数组，该数组是由分隔符分隔开的子串构成的。但是，这个分隔符是不返回的。实现下面的新方法，方法返回字符串数组，这个数组由用匹配字符分隔开的子串构成，子串也包括匹配字符。

public static String[] split(String s, String regex)

例如，split("ab#12#453", "#")，会返回 ab、#、12、# 和 453 构成的 String 数组，而 split("a?b?gf#e", "[?#]")会返回 a、？ 、b、?、gf、#和 e 构成的字符串数组。

26. （计算器）修改教材程序清单 7-9（参见教材 P_{233}），接收一个字符串的表达式，其中操作符和操作数由 0 到多个空格隔开。例如，3+4 和 3 + 4 都是可以接受的表达式。下面是一个运行示例：



27. （实现 StringBuilder 类）在 Java 库中提供了 StringBuilder 类。给出你自己对下面方法的实现（将新类命名为 MyStringBuilderl):

public MyStringBuilderl(String s);
public MyStringBuilderl append(MyStringBuilderl s);
public MyStringBuilderl append(int i);
public int length();
public char charAt(int index);
public MyStringBuilderl toLowerCase();
public MyStringBuilderl substring(int begin, int end);
public String toString();

28. （实现 StringBuilder 类）在 Java 库中提供了 StringBuilder 类。给出你自己对下面方法的实现（将新类命名为 MyStringBirilder2):

public MyStringBui1der2();

```
public MyStringBui1der2(char[] chars);
public MyStringBuilder2(String s);
public MyStringBuilder2 insert(int offset, MyStringBui1der2 s);
public MyStringBuilder2 reverse();
public MyStringBuilder2 substring(int begin);
public MyStringBuilder2 toUpperCase();
```

实验 6 类的继承性和多态性

1. （三角形类 Triangle）设计一个名为 Triangle 的类来扩展 GeometricObject 类（参见教材 P_{349}）。该类包括：
   - 三个名为 sidel、side2 和 side3 的 double 数据域表示这个三角形的三条边，它们的默认值是 1.0。
   - 一个无参构造方法创建默认的三角形。
   - 一个能创建带指定 sidel、side2 和 side3 的三角形的构造方法。
   - 所有三个数据域的访问器方法。
   - —个名为 getArea（）的方法返回这个三角形的面积。
   - 一个名为 getPerimeter（）的方法返回这个三角形的周长。
   - —个名为 toString（）的方法返回这个三角形的字符串描述。

   计算三角形面积的公式参见编程练习题 31。toString（）方法的实现如下所示：

```
return "Triangle: side1 = " + side1 + " side2 = " + side2 +
  " side3 = " + side3;
```

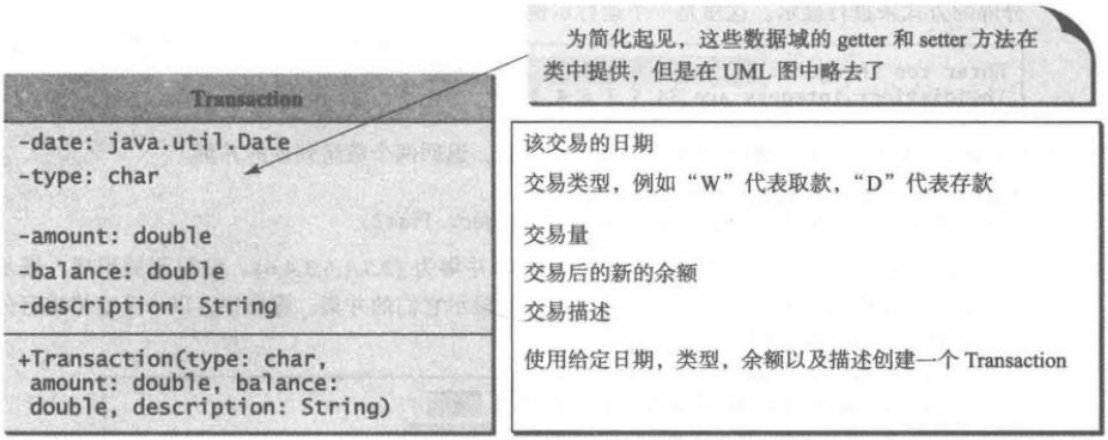
   实现类 Triangle 和 GeometricObject。编写一个测试程序，提示用户输入三角形的三条边、颜色以及一个 Boolean 值表明该三角形是否填充。程序应该使用输入创建一个具有这些边并设置 color 和 filled 属性的三角形。程序应该显示面积、边长、颜色以及表明是否填充的真或者假的值。
2. （Person、Student、Employee、Faculty 和 Staff 类）设计一个名为 Person 的类和它的两个名为 Student 和 Employee 的子类。Employee 类又有子类：教员类 Faculty 和职员类 Staff。每个人都有姓名、地址、电话号码和电子邮件地址。学生有班级状态（大一、大二、大三或大四）。将这些状态定义为常量。一个雇员涉及办公室、工资和受聘日期。使用编程练习题 299 中定义的 MyDate 类为受聘日期创建一个对象。教员有办公时间和级别。职员有职务称号。覆盖每个类中的 toString 方法，显示相应的类别名字和人名。实现这些类，并编写一个测试程序，创建 Person、Student、Employee、Faculty 和 Staff，并且调用它们的 toString（）方法。
3. （账户类 Account 的子类）在编程练习题 9.7（参见教材 P_{306}）中定义了一个 Account 类来建模一个银行账户。一个账户有账号、余额、年利率、开户日期等属性，以及存款和取款等方法。创建两个检测支票账户（checking account）和储蓄账户（saving account）的子类。支票账户有一个透支限定额，但储蓄账户不能透支。实现这些类，并编写一个测试程序，创建 Account、SavingsAccount 和 CheckingAccount 的对象，然后调用它们的 toString（）方法。
4. （ArrayList 的最大元素）编写以下方法，返回一个整数 ArrayList 的最大值。如果列表为 null 或者列表的大小为 0，则返回 null 值。

   public static Integer max（ArrayList<Integer> list）

   编写一个测试程序，提示用户输入一个以 0 结尾的数值序列，调用该方法返回输入的最大数值。
5. （课程类 Course）重写程序清单 10-6（参见教材 P_{319}）中的 Course 类，使用 ArrayList 代替数组来存储学生，不应该改变 Course 类的原始合约（即，构造方法和方法的定义都不应该改变，但私有的成员可以改变）。
6. （使用 ArrayList）编写程序，创建一个 ArrayList，然后向这个列表中添加一个 Loan 对象、—个 Date 对象、一个字符串和一个 Circle 对象，然后使用循环调用对象的 toString

（）方法，来显示列表中所有的元素。

7. （打乱 ArrayList）编写以下方法，打乱一个整数 ArrayList 中的元素。
   public static void shuff1e（ArrayList<Integer> list）
8. （新的 Account 类）编程练习题 9.7（参见教材 P_{306}）中给出了一个 Account 类，如下设计一个新的 Account 类：
   - 添加一个 String 类型的新数据域 name 来存储客户的名字。
   - 添加一个新的构造方法，该方法创建一个具有指定名字、id 和收支额的账户。
   - 添加一个名为 transactions 的 ArrayList 类型的新数据域用于为账户存储交易。每笔交易都是一个 Transaction 类的实例。Transaction 类的定义如下图所示。
   - 修改 withdraw 和 deposit 方法，向 transactions 数组线性表添加一笔交易。
   - 其他所有属性和方法都和编程练习题 262 中的一样。



编写一个测试程序，创建一个年利率为 1.5%、收支额为 1000、id 为 1122 而名字为 George 的 Account。向该账户存入 30 美元、40 美元和 50 美元并从该账户中取出 5 美元、4 美元和 2 美元。打印账户清单，显示账户持有者名字、利率、收支额和所有的交易。

9. （最大的行和列）编写程序，随机将 0 和 1 填入一个 n*n 的矩阵，打印该矩阵，并且找出具有最多 1 的行和列。提示：使用两个 ArrayList 来存储具有最多 1 的行和列的下标。这里是程序的一个运行示例：

```
Enter the array size n: 4 ↵Enter
The random array is
0011
0011
1101
1010
The largest row index: 2
The largest column index: 2, 3
```

10. （利用继承实现 MyStack）在程序清单 11-10（参见教材 P_{375}）中，MyStack 是用组合实现的。扩展 ArrayList 创建一个新的栈类。实现 MyStack 类，并编写一个测试程序，提示用户输入五个字符串，然后以逆序显示这些字符串。
11. （对 ArrayList 排序）编写以下方法，对一个数值的 ArrayList 进行排序：
    public static void sort（ArrayList<Integer> list）
    编写测试程序，提示用户输入 5 个数字，将其存储在一个数组列表中，并且以升序进行显示。
12. （对 ArrayList 求和）编写以下方法，返回 ArrayList 中所有数字的和：
    public static double sum（ArrayList<Double> list）

编写测试程序，提示用户输入 5 个数字，将其存储在一个数组列表中，并且显示它们的和。

13. （去掉重复元素）使用下面的方法头编写方法，从一个整数的数组列表中去掉重复元素：
public static void removeDuplicate（ArrayList<Integer> list）
编写测试程序，提示用户输入 10 个整数到列表中，显示其中不同的整数，并以一个空格分隔的方式来进行显示。这里是一个运行示例：

```
Enter ten integers: 34 5 3 5 6 4 33 2 2 4 Enter
The distinct integers are 34 5 3 6 4 33 2
```

14. （结合两个列表）使用下面的方法头编写一个方法，返回两个数组列表的并集。
public static ArrayList<Integer> union（ArrayList<Integer> listl，ArrayList<Integer> list2）
例如，两个数组列表{2，3，1，5}和{3，4，6}的并集为{2，3，1，5，3，4，6}。编写测试程序，提示用户输入两个列表，每个列表有 5 个整数，然后显示它们的并集。输出中，以一个空格进行分隔。这里是一个运行示例：

```
Enter five integers for list1: 3 5 45 4 3 Enter
Enter five integers for list2: 33 51 5 4 13 Enter
The combined list is 3 5 45 4 3 33 51 5 4 13
```

15. （凸多边形面积）如果一个多边形中连接任意两个顶点的线段都包含在多边形中，则称为凸多边形。编写一个程序，提示用户输入一个凸多边形中的顶点数，并顺时针输入点，然后程序显示多边形的面积信息。这里是一个程序的运行示例：

```
Enter the number of the points: 7 Enter
Enter the coordinates of the points:
  -12 0 -8.5 10 0 11.4 5.5 7.8 6 -5.5 0 -7 -3.5 -13.5 Enter
The total area is 292.575
```

16. （加法测试）重写程序清单 5-1（参见教材 P_{135}），如果用户重复输入了相同的答案，则给出用户警告。提示：使用一个数组列表来存储答案。这里是一个运行示例：

```
What is 5 + 9? 12 Enter
Wrong answer. Try again. What is 5 + 9? 34 Enter
Wrong answer. Try again. What is 5 + 9? 12 Enter
You already entered 12
Wrong answer. Try again. What is 5 + 9? 14 Enter
You got it!
```

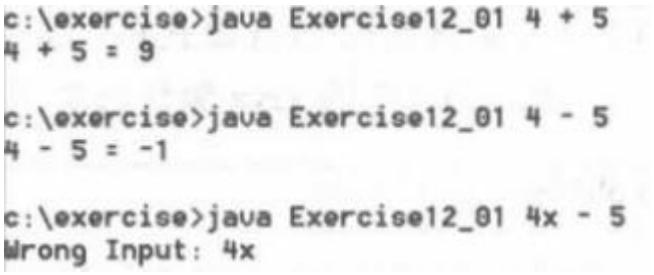
17. （代数：完全平方）编写一个程序，提示用户输入一个整数 m，然后找到最小的整数 n，使得 m*n 是一个完全平方。提示：存储所有 m 的最小因子到一个数组列表，则 n 是列表中出现奇数次的因子的乘积。例如，考虑 m=90 的情况，保存因子 2，3，3，5 到一个数组列表中。列表中 2 和 5 出现了奇数次数，因此，n 是 10。这里是运行示例：

```
Enter an integer m: 1500
The smallest number n for m * n to be a perfect square is 15
m * n is 22500
```

```
Enter an integer m: 63
The smallest number n for m * n to be a perfect square is 7
m * n is 441
```

实验 7 异常处理

1. （NumberFormatException 异常）程序清单 7-9（参见教材 P_{233}）是一个简单的命令行计算器。注意，如果某个操作数是非数值的，程序就会中止。编写一个程序，利用异常处理器来处理非数值操作数；然后编写另一个不使用异常处理器的程序，达到相同的目的。程序在退出之前应该显示一条消息，通知用户发生了操作数类型错误，如下图所示。



2. （InputMismatchException 异常）编写一个程序，提示用户读取两个整数，然后显示它们的和。程序应该在输入不正确时提示用户再次读取数字。
3. （ArraylndexOutBoundsExcepfion 异常）编写一个满足下面要求的程序：
   - 创建一个由 100 个随机选取的整数构成的数组。
   - 提示用户输入数组的下标，然后显示对应的元素值。如果指定的下标越界，就显示消息 Out of Bounds。
4. （IllegalArgumentException 异常）修改程序清单 10-2（参见教材 P_{311}）中的 Loan 类，如果贷款总额、利率、年数小于或等于零，则抛出 IllegalArgumentException 异常。
5. （IllegalTriangleException 异常）编程练习题 269 定义了带三条边的 Triangle 类。在三角形中，任意两边之和总大于第三边，三角形类 Triangle 必须遵从这一规则。创建一个 IllegalTriangleException 类，然后修改 Triangle 类的构造方法，如果创建的三角形的边违反了这一规则，抛出一个 IllegalTriangleException 对象，如下所示：

```
/**Constructatrianglewiththespecifiedsides*/
public Triangle（double sidel，double side2，double side3）
        throws IllegalTriangleException{
            //Implement it
}
```

6. （NumberFormatExceprion 异常）程序清单 6-8（参见教材 P_{184}）实现了 hexToDec（StringhexString）方法，它将一个十六进制字符串转换为一个十进制数。实现这个 hexToDec 方法，在字符串不是一个十六进制字符串时抛出 NumberFormatException 异常。
7. （NumberFormatException 异常）编写 bin2Dec（StringbinaryString）方法，将一个二进制字符串转换为一个十进制数。实现 bin2Dec 方法，在字符串不是一个二进制字符串时抛出 NumberFormatException 异常。
8. （HexFormatException 异常）编程练习题 319 实现 hex2Dec 方法，在字符串不是一个十六进制字符串时抛出 NumberFormatException 异常。定义一个名为 HexFormatException 的自定义异常。实现 hex2Dec 方法，在字符串不是一个十六进制字符串时抛出 HexFormatException 异常。
9. （BinaryFormatException 异常）编程练习题 320 实现 bin2Dec 方法，在字符串不是一个二进制字符串时抛出 BinaryFormatException 异常。定义一个名为 BianryFormatException 的自定义异常。实现 bin2Dec 方法，在字符串不是一个二进制字符串时抛出

BinaryFormatException 异常。

10. （OutOfMemoryError 错误）编写一个程序，它能导致 JVM 抛出一个 OutOfMemoryError，然后捕获和处理这个错误。

实验 8 抽象类和接口的应用

1. （三角形类）设计一个扩展自抽象类 GeometricObject 的新的 Triangle 类。实现 Triangle 类，并编写一个测试程序，提示用户输入三角形的三条边、一种颜色以及一个表明该三角形是否填充的布尔值。程序应该根据用户的输入，使用这些边以及颜色和是否填充的信息，创建一个 Triangle 对象。程序应该显示面积、周长、颜色以及真或者假来表明是否被填充。
2. （打乱 ArrayList）编写以下方法，打乱 ArrayList 里面保存的数字。
   public static void shuff1e（ArrayList<Number> list）
3. （排序 ArrayList）编写以下方法，对 ArrayList 里面保存的数字进行排序。
   public static void sort（ArrayList<Number> list）
4. （显示日历）重写程序清单 6-12（参见教材 P_{194}）中的 PrintCalendar 类，使用 Calendar 和 GregorianCalendar 类显示一个给定月份的日历。你的程序从命令行得到月份和年份的输入，例如：
   java Exercisel3_04 5 2016
   这个输入会显示如下图中的日历。也可以不输入年份来运行程序。这种情况下，年份就是当前年份。如果不指定月份和年份来运行程序，那么就是指当前月份。

```
c:\exercise>java Exercise13_04 5 2016
          May, 2016
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
   1   2   3   4   5   6   7
   8   9  10  11  12  13  14
  15  16  17  18  19  20  21
  22  23  24  25  26  27  28
  29  30  31
```

5. （将 GeometricObject 类变成可比较的）修改 GeometricObject 类以实现 Comparable 接口，并且在 GeometricObject 类中定义一个静态的求两个 GeometricObject 对象中较大者的 max 方法。实现这个新的 GeometricObject 类，并编写一个测试程序，使用 max 方法求两个圆中的较大者和两个矩形中的较大者。
6. （ComparableCircle 类）创建名为 ComparableCircle 的类，它继承自 Circle 类，并实现 Comparable 接口。实现 compareTo 方法，使其根据面积比较两个圆。编写一个测试程序求出 ComparableCircle 对象的两个实例中的较大者。
7. （可着色接口 Colorable）设计一个名为 Colorable 的接口，其中有名为 howToColor（）的 void 方法。可着色对象的每个类必须实现 Colorable 接口。设计一个名为 Square 的类，继承自 GeometricObject 类并实现 Colorable 接口。实现 howToColor 方法，显示一个消息 Color all four sides（给所有的四条边着色）。编写一个测试程序，创建有五个 GeometricObject 对象的数组。对于数组中的每个对象而言，如果对象是可着色的，那就调用 howToColor 方法。
8. （修改 MyStack 类）重写程序清单 11-10（参见教材 P_{375}）中的 MyStack 类，执行 list 域的深度复制。
9. （将 Circle 类改成可比较的）改写程序清单 13-2（参见教材 P_{426}）中的 Circle 类，它继承自 GeometricObject 类并实现 Comparable 接口。覆盖 Object 类中的 equals 方法。当两个 Circle 对象半径相等时，则这两个 Circle 对象是相同的。

10. （将 Rectangle 类变成可比较的）改写程序清单 13-3（参见教材 P_{426}）的 Rectangle 类，它继承自 GeometricObject 类并实现 Comparable 接口。覆盖 Object 类中的 equals 方法。当两个 Rectangle 对象面积相同时，则这两个对象是相同的。
11. （八边形类 Octagon）编写一个名为 Octagon 的类，它继承自 GeometricObject 类并实现 Comparable 和 Cloneable 接口。假设八边形八条边的边长都相等，它的面积可以使用下面的公式计算：

$$面积 = (2 + 4 / \sqrt{2}) \times 边长 \times 边长$$

编写一个测试程序，创建一个边长值为 5 的 Octagon 对象，然后显示它的面积和周长。使用 clone 方法创建一个新对象，并使用 compareTo 方法比较这两个对象。

12. （求几何对象的面积之和）编写一个方法，求数组中所有几何对象的面积之和。方法签名如下：

public static double sumArea（GeometricObject[] a）

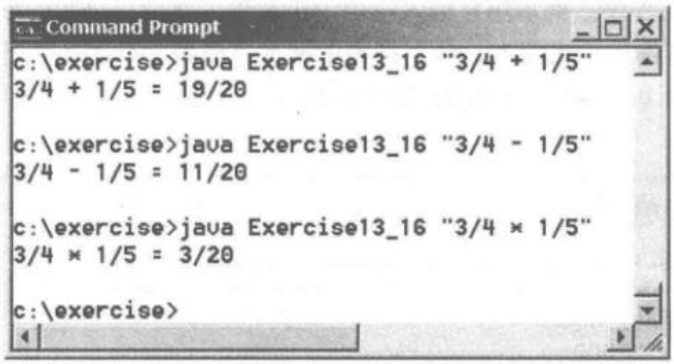
编写测试程序，创建四个对象（两个圆和两个矩形）的数组，然后使用 sumArea 方法求它们的总面积。

13. （使得 Course 类可复制）重写程序清单 10-6（参见教材 P_{319}）中的 Course 类，增加一个 clone 方法，执行 students 域上的深度复制。
14. （演示封装的好处）使用新的分子分母的内部表达改写程序清单 13-13（参见教材 P_{449}）中的 Rational 类。创建有两个整数的数组，如下所示：

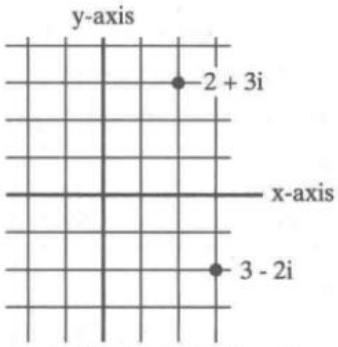
private long[] r = new 1ong[2]:

使用 r[0]表示分子，使用 r[l]表示分母。在 Rational 类中的方法签名没有改变，因此，无须重新编译，前一个 Rational 类的客户端应用程序可以继续使用这个新的 Rational 类。

15. （在 Rarional 类中使用 BigInteger）使用 BigInteger 表示分子和分母，重新设计和实现程序清单 13-13（参见教材 P_{449}）中的 Rational 类。
16. （创建一个有理数的计算器）编写一个类似于程序清单 7-9（参见教材 P_{233}）的程序。这里不使用整数，而是使用有理数，如下图 a 所示。需要使用在教材 10.10.3 节（参见教材 P_{329}）中介绍的 String 类中的 split 方法来获取分子字符串和分母字符串，并使用 Integer.parseInt 方法将字符串转换为整数。



a）程序从命令行得到三个参数（操作数 1、操作符、操作数 2），显示该表达式以及算数运算的结果



b）复数可以解释为一个平面上的点

17. （数学：Complex 类）一个复数是一个形式为 a+bi 的数，这里的 a 和 b 都是实数，i 是 $\sqrt{-1}$的平方根。数字 a 和 b 分别称为复数的实部和虚部。可以使用下面的公式完成复数的加、减、乘、除：

$$a+bi+c+di=(a+c)+(b+d)i$$
$$a+bi-(c+di)=(a-c)+(b-d)i$$
$$(a+bi)*(c+di)=(ac-bd)+(bc+ad)i$$
$$(a+bi)/(c+di)=(ac+bd)/(c^2+d^2)+(bc-ad)i/(c^2+d^2)$$

还可以使用下面的公式得到复数的绝对值：

$$|a+bi|=\sqrt{a^2+b^2}$$

（复数可以解释为一个平面上的点，将值作为该点的坐标。复数的绝对值是该点到原点的距离，如上图 b 所示。）

设计一个名为 Complex 的复数来表示复数以及完成复数运算的 add、substract、multiply、divide 和 abs 方法，并且覆盖 toStrlng 方法以返回一个表示复数的字符串。方法 toString 返回字符串 a+bi。如果 b 是 0，那么它只返回 a。Complex 类应该也实现 Cloneable 接口。提供三个构造方法 Complex（a，b）、Complex（a）和 Complex（）。Comp1ex（）创建数字 0 的 Complex 对象，而 Complex（a）创建一个 b 为 0 的 Complex 对象。还提供 getRealPart（）和 getImaginaryPart（）方法以分别返回复数的实部和虚部。编写一个测试程序，提示用户输入两个复数，然后显示它们做加、减、乘、除之后的结果。下面是运行示例：

```
Enter the first complex number:  3.5 5.5 Enter
Enter the second complex number:  -3.5 1 Enter
(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i
(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i
(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -13.75i
(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i
|(3.5 + 5.5i)| = 6.519202405202649
```

18. （使用 Rational 类）编写程序，使用 Rational 类计算下面的求和数列：

$$\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{98}{99}+\frac{99}{100}$$

你将会发现输出是不正确的，因为整数溢出（太大了）。为了解决这个问题，参见编程练习题 338。

19. （将十进制数转化为分数）编写一个程序，提示用户输入一个十进制数，然后以分数的形式显示该数字。提示：将十进制数以字符串的形式输入，从字符串中抽取其整数部分和小数部分，然后运用编程练习题 338 中使用 Biglnteger 实现的 Rational 类，来获得该十进制数的有理数。下面是一些运行示例：

```
Enter a decimal number: 3.25 Enter
The fraction number is 13/4
```

```
Enter a decimal number: -0.45452 Enter
The fraction number is -11363/25000
```

20. （数学：求解二元方程）重写编程练习题 35，如果行列式小于 0，则使用编程练习题 340

中的 Complex 类来得到虚根。下面是一些运行示例：

```
Enter a, b, c: 1 3 1 ↵Enter
The roots are -0.381966 and -2.61803
```

```
Enter a, b, c: 1 2 1 ↵Enter
The root is -1
```

```
Enter a, b, c: 1 2 3 ↵Enter
The roots are -1.0 + 1.4142i and -1.0 + -1.4142i
```

21. （代数：顶点式方程）抛物线方程可以表达为标准形式（ax^2+bx+c）或者顶点式（$a(x-h)^2+k$）。编写一个程序，提示用户输入标准形式下的整数 a、b 和 c 值，显示顶点式下面的 h 和 k 值。下面是一些运行示例：

```
Enter a, b, c: 1 3 1 ↵Enter
h is -3/2 k is -5/4
```

```
Enter a, b, c: 2 3 4 ↵Enter
h is -3/4 k is 23/8
```