

置都会显示在控制台上。然而，查找逻辑错误就很富有挑战性。

逻辑错误也称为臭虫（bug）。查找和改正错误的过程称为调试（debugging）。调试的一般途径是采用各种方法逐步缩小程序中 bug 所在的范围。可以手工跟踪（hand trace）程序（即通过读程序找错误），也可以插入打印语句，显示变量的值或程序的执行流程。这种方法适用于短小、简单的程序。对于庞大、复杂的程序，最有效的调试方法还是使用调试工具。

JDK 包含了一个命令行调试器 jdb，结合一个类名来调用该命令。Jdb 本身也是一个 Java 程序，运行自身的一个 Java 解释器的拷贝。所有的 Java IDE 工具，比如 Eclipse 和 NetBeans 包含集成的调试器。调试器应用让你可以跟踪一个程序的执行。它们因系统而不同，但是都支持以下有用的特征中的多数。

- 一次执行一条语句：调试器允许你一次执行一条语句，从而可以看到每条语句的效果。
- 跟踪进入或者一步运行过一个方法：如果一个方法正在被执行，你可以让调试器跟踪进入方法内部，并且一次执行方法里面的一条语句，或者你也可以让调试器一步运行过整个方法。如果你知道方法是可行的，你应该一次运行过整个的方法。比如，通常都会一步运行过系统提供的方法，比如 `System.out.println`。
- 设置断点：你也可以在一条特定的语句上面设置断点。当遇到一个断点时，你的程序将暂停。你可以设置任意多的断点。当你知道程序错误从什么地方可能开始的时候，断点特别有用。你可以将断点设置在那条语句上，让程序先执行到断点处。
- 显示变量：调试器让你选择多个变量并且显示它们的值。当你跟踪一个程序的时候，变量的内容持续更新。
- 显示调用堆栈：调试器让你跟踪所有的方法调用。当你需要看到程序执行流程的宏观图景的时候，这个特征非常有用。
- 修改变量：一些调试器允许你在调试的过程中修改变量的值。当你希望用不同的示例来测试程序，而又不希望离开调试器的时候，这是非常方便的。

✎ 提示：如果你使用诸如 Eclipse 或者 NetBeans 之类的 IDE，请参考配套网站上面的补充材料 II.C 和 II.E。补充材料给出如何使用调试器来追踪程序，以及调试过程是如何帮助你有效学习 Java 的。

关键术语

Boolean expression（布尔表达式）

Boolean data type（boolean 数据类型）

Boolean value（布尔值）

conditional operator（条件操作符）

dangling-else ambiguity（悬空 else 歧义）

debugging（调试）

fall-through behavior（落空行为）

flowchart（流程图）

lazy operator（懒惰操作符）

operator associativity（操作符结合规则）

operator precedence（操作符优先级）

selection statement（选择语句）

short-circuit evaluation（短路运算）

本章小结


1. boolean 类型变量可以存储值 true 或 false。
2. 关系操作符（<、<=、==、!=、>、>=）产生一个布尔值。


3. 选择语句用于可选择的动作路径的编程。选择语句有以下几种类型：单分支 if 语句、双分支 if-else 语句、嵌套 if 语句、多分支 if-else 语句、switch 语句和条件表达式。
4. 各种 if 语句都是基于布尔表达式来控制决定的。根据表达式的值是 true 或 false，这些语句选择两种可能路径中的一种。
5. 布尔操作符 &&、||、| 和 ^ 对布尔值和布尔变量进行计算。
6. 当对 p1&p2 求值时，Java 先求 p1 的值，如果 p1 为 true，再对 p2 求值；如果 p1 为 false，就不再对 p2 求值。当对 p1||p2 求值时，Java 先求 p1 的值，如果 p1 为 false，再对 p2 求值；如果 p1 为 true，就不再对 p2 求值。因此，&& 也称为条件与操作符或短路与操作符，而 || 也称为条件或操作符或短路或操作符。
7. switch 语句根据 char、byte、short、int 或者 String 类型的 switch 表达式来进行控制决定。
8. 在 switch 语句中，关键字 break 是可选的，但它通常用在每个分支的结尾，以中止执行 switch 语句的剩余部分。如果没有出现 break 语句，则执行接下来的 case 语句。
9. 表达式中的操作符按照括号、操作符优先级以及操作符结合规则所确定的次序进行求值。
10. 括号用于强制求值的顺序以任何顺序进行。
11. 具有更高级优先权的操作符更早地进行操作。对于同样优先级的操作符，它们的结合规则确定操作的顺序。
12. 除开赋值操作符的所有二元操作符都是左结合的，赋值操作符是右结合的。

测试题

在线回答本章测试题，地址为 www.cs.armstrong.edu/liang/intro10e/quiz.html。

编程练习题

 **教学提示** 对于每一个练习题，都应在编码之前仔细地分析问题的需求及设计解题策略。

 **调试提示** 在寻求帮助之前，将程序读和解释给自己听。然后通过手动使用几个具有代表性的输入，或者使用某个 IDE 调试器来跟踪程序。通过调试自己编程中的错误，来学习如何编程。


3.2 节

*3.1（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：

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

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“The equation has no real roots”（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。

```
Enter a, b, c: 1.0 3 1
The equation has two roots -0.381966 and -2.61803
```

```
Enter a, b, c: 1 2.0 1
The equation has one root -1
```

```
Enter a, b, c: 1 2 3
The equation has no real roots
```

3.2 (游戏: 三个数的加法) 程序清单 3-1 中的程序产生两个整数, 并提示用户输入这两个整数的和。

修改该程序使之能产生三个一位整数, 然后提示用户输入这三个整数的和。

*3.3 (代数: 求解 2×2 线性方程) 可以使用编程练习题 1.13 中给出的 Cramer 规则解线性方程组:

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

编写程序, 提示用户输入 a, b, c, d, e 和 f , 然后显示结果。如果 $ad - bc$ 为 0, 报告消息 “The equation has no solution” (方程式无解)。

```
Enter a, b, c, d, e, f: 9.0 4.0 3.0 -5.0 -6.0 -21.0 Enter
x is -2.0 and y is 3.0
```

```
Enter a, b, c, d, e, f: 1.0 2.0 2.0 4.0 4.0 5.0 Enter
The equation has no solution
```

**3.4 (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, ..., 12 显示相应的英文月份: January, February, ..., December。

*3.5 (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:

```
Enter today's day: 1 Enter
Enter the number of days elapsed since today: 3 Enter
Today is Monday and the future day is Thursday
```

```
Enter today's day: 0 Enter
Enter the number of days elapsed since today: 31 Enter
Today is Sunday and the future day is Wednesday
```

*3.6 (医疗应用程序: BMI) 修改程序清单 3-4, 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
```

3.7 (财务应用程序: 整钱兑零) 修改程序清单 2-10, 使之只显示非零的币值单位, 用单词的单数形式显示一个单位, 例如 1 dollar and 1 penny (1 美元和 1 美分); 用单词的复数形式显示多于一个单位的值, 例如 2 dollars and 3 pennies (2 美元和 3 美分)。

*3.8 (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。

**3.9 (商业: 检查 ISBN-10) ISBN-10 (国际标准书号) 以前是一个 10 位整数 $d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10}$, 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 ISBN-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。下面是一个运行示例:

```
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
```

- 3.10 (游戏: 加法测验) 程序清单 3-3 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。

3.8 ~ 3.16 节

- *3.11 (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)。
- 3.12 (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
```

- *3.13 (财务应用程序: 计算税款) 程序清单 3-5 给出了计算单身登记人税款的源代码。将程序清单 3-5 补充完整, 从而计算所有登记的婚姻状态的税款。
- 3.14 (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。
- **3.15 (游戏: 彩票) 修改程序清单 3-8, 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 10 000 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 3 000 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 1 000 美元。
- 3.16 (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。
- *3.17 (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. You won
```

```
scissor (0), rock (1), paper (2): 2
The computer is paper. You are paper too. It is a draw
```

- *3.18 (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)。

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

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 cannot be shipped”。

****3.19** (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。

***3.20** (科学：风寒温度) 编程练习题 2.17 给出计算风寒温度的公式。这个公式适用于温度在华氏 -58° 到 41° 之间，并且风速大于或等于 2 的情况。编写一个程序，提示用户输入一个温度值和一个风速值。如果输入值是合法的，那么显示风寒温度，否则显示一条消息，表明温度或风速是不合法数值。

综合题

****3.21** (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度的 13 和 14 月。
- j 是世纪数 (即 $\left\lfloor \frac{\text{year}}{100} \right\rfloor$)。
- k 是该世纪的第几年 (即 $\text{year} \% 100$)。


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
```

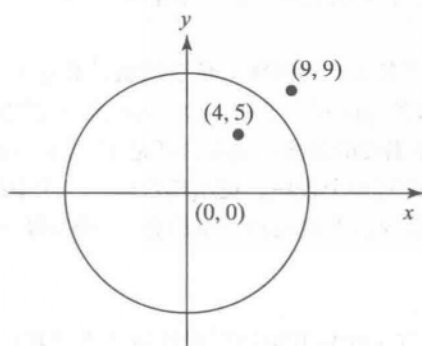
 **提示：**一月和二月在这个公式里是用 13 和 14 表示的。所以需要将由用户输入的月份 1 转换为 13，将由用户输入的月份 2 转换为 14，同时将年份改为前一年。

****3.22** (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, 5) 是圆内的一点，而 (9, 9) 是圆外的一点，如图 3-7a 所示。

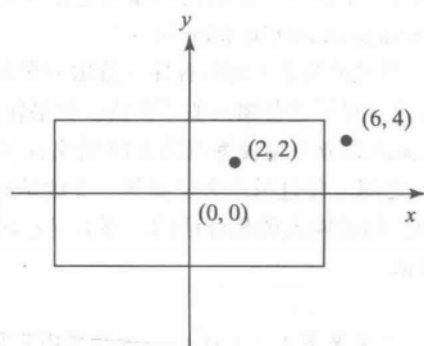
 **提示：**如果一个点到 $(0, 0)$ 的距离小于或等于 10，那么该点就在圆内，计算距离的公式是 $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ ，使用各种情况来测试你的程序。以下是两个运行示例。

```
Enter a point with two coordinates: 4 5 Enter
Point (4.0, 5.0) is in the circle
```

```
Enter a point with two coordinates: 9 9 Enter
Point (9.0, 9.0) is not in the circle
```



a) 圆内和圆外的点



b) 矩形外和矩形内的点

图 3-7

****3.23** (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, 如图 3-7b 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 Enter
Point (2.0, 2.0) is in the rectangle
```

```
Enter a point with two coordinates: 6 4 Enter
Point (6.0, 4.0) is not in the rectangle
```

****3.24** (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
```

***3.25** (几何: 交点) 第一条直线上面的两个点是 (x_1, y_1) 和 (x_2, y_2) , 第二条直线的两个点是 (x_3, y_3) 和 (x_4, y_4) , 如图 3-8a、图 3-8b 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 3.3)。如果方程无解, 则两条直线平行 (图 3-8c)。

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 x1, y1, x2, y2, x3, y3, x4, y4: 2 2 5 -1.0 4.0 2.0 -1.0 -2.0 Enter
The intersecting point is at (2.88889, 1.1111)
```

```
Enter x1, y1, x2, y2, x3, y3, x4, y4: 2 2 7 6.0 4.0 2.0 -1.0 -2.0 Enter
The two lines are parallel
```

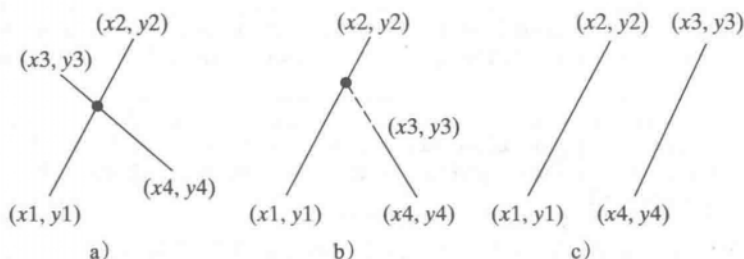
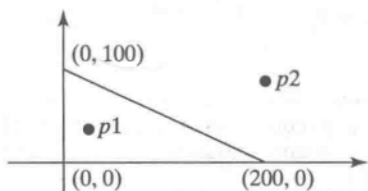


图 3-8 两条直线相交 (a 和 b), 两条直线平行 (c)

3.26 (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
```

**3.27 (几何: 点是否在三角形内?) 假设一个直角三角形放在一个平面上, 如下图所示。直角点在 (0, 0) 处, 其他两个点分别在 (200, 0) 和 (0, 100) 处。编写程序, 提示用户输入一个点的 x 坐标和 y 坐标, 然后判定这个点是否在该三角形内。下面是运行示例:



```
Enter a point's x- and y-coordinates: 100.5 25.5
The point is in the triangle
```

```
Enter a point's x- and y-coordinates: 100.5 50.5
The point is not in the triangle
```

**3.28 (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, 如图 3-9 所示。

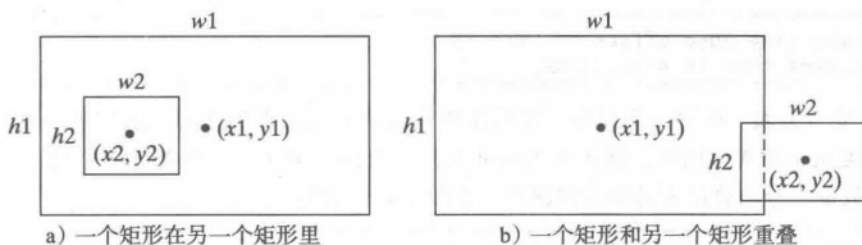


图 3-9

下面是运行示例:


```
Enter r1's center x-, y-coordinates, width, and height: 2.5 4 2.5 43
Enter r2's center x-, y-coordinates, width, and height: 1.5 5 0.5 3
r2 is inside r1
```



```
Enter r1's center x-, y-coordinates, width, and height: 1 2 3 5.5 --Enter
Enter r2's center x-, y-coordinates, width, and height: 3 4 4.5 5 --Enter
r2 overlaps r1
```

```
Enter r1's center x-, y-coordinates, width, and height: 1 2 3 3 --Enter
Enter r2's center x-, y-coordinates, width, and height: 40 45 3 2 --Enter
r2 does not overlap r1
```

****3.29 (几何: 两个圆)** 编写程序, 提示用户输入两个圆的中心坐标和各自的半径值, 然后决定第二个圆是在第一个圆内, 还是和第一个圆重叠, 如图 3-10 所示。

 **提示:** 如果两个圆心的距离 $\leq |r1 - r2|$, 就认为 circle2 在 circle1 内; 如果两个圆心的距离 $\leq r1 + r2$, 就认为 circle2 和 circle1 重叠。

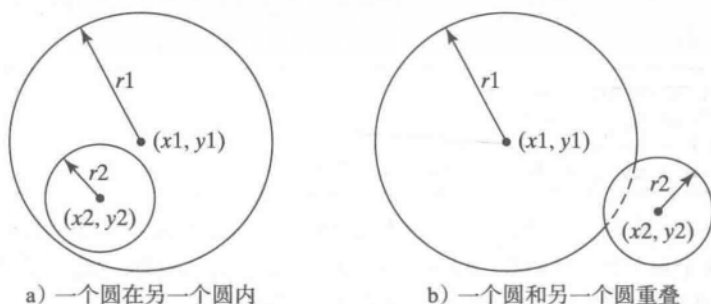


图 3-10

下面是运行示例:

```
Enter circle1's center x-, y-coordinates, and radius: 0.5 5.1 13 --Enter
Enter circle2's center x-, y-coordinates, and radius: 1 1.7 4.5 --Enter
circle2 is inside circle1
```

```
Enter circle1's center x-, y-coordinates, and radius: 3.4 5.7 5.5 --Enter
Enter circle2's center x-, y-coordinates, and radius: 6.7 3.5 3 --Enter
circle2 overlaps circle1
```

```
Enter circle1's center x-, y-coordinates, and radius: 3.4 5.5 1 --Enter
Enter circle2's center x-, y-coordinates, and radius: 5.5 7.2 1 --Enter
circle2 does not overlap circle1
```

***3.30 (当前时间)** 修改编程练习题 2.8, 以 12 小时制显示小时数。这里是一个运行示例:

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

***3.31 (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
Enter 0 to convert dollars to RMB and 1 vice versa: 0 --Enter
Enter the dollar amount: 100 --Enter
$100.0 is 681.0 yuan
```

```
Enter the exchange rate from dollars to RMB: 6.81 --Enter
Enter 0 to convert dollars to RMB and 1 vice versa: 1 --Enter
Enter the RMB amount: 10000 --Enter
10000.0 yuan is $1468.43
```



```
Enter the exchange rate from dollars to RMB: 6.81 ↵ Enter
Enter 0 to convert dollars to RMB and 1 vice versa: 5 ↵ Enter
Incorrect input
```

*3.32 (几何: 点的位置) 给定一个从点 $p_0(x_0, y_0)$ 到 $p_1(x_1, y_1)$ 的有向线段, 可以使用下面的条件来确定点 $p_2(x_2, y_2)$ 是在线段的左侧、右侧, 或者在该直线上 (见图 3-11):

$$(x_1 - x_0) \times (y_2 - y_0) - (x_2 - x_0) \times (y_1 - y_0) \begin{cases} > 0 & p_2 \text{ 在线段的左侧} \\ = 0 & p_2 \text{ 在该线段上} \\ < 0 & p_2 \text{ 在线段的右侧} \end{cases}$$

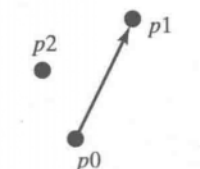
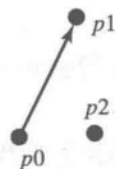
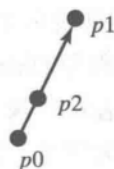
a) p_2 在线段的左侧b) p_2 在线段的右侧c) p_2 在该线段上

图 3-11

编写一个程序, 提示用户输入三个点 p_0 、 p_1 和 p_2 , 显示 p_2 是否在从 p_0 到 p_1 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 ↵ Enter
(-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 ↵ Enter
(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 ↵ Enter
(5.0, 2.5) is on the right side of the line from (3.4, 2.0) to (6.5, 9.5)
```

*3.33 (金融: 比较成本) 假设你要通过两种不同的包裹运输大米。你将乐于编写一个程序来比较成本, 该程序提示用户输入每个包裹的重量和价格, 然后显示具有更好价格的包裹。下面是一个运行示例:

```
Enter weight and price for package 1: 50 24.59 ↵ Enter
Enter weight and price for package 2: 25 11.99 ↵ Enter
Package 2 has a better price.
```

```
Enter weight and price for package 1: 50 25 ↵ Enter
Enter weight and price for package 2: 25 12.5 ↵ Enter
Two packages have the same price.
```

*3.34 (几何: 线段上的点) 编程练习题 3.32 显示了如何测试一个点是否在一个无限长的直线上。修改编程练习题 3.32, 测试一个点是否在一个线段上。编写一个程序, 提示用户输入三个点 p_0 、 p_1 和 p_2 , 显示 p_2 是否在从 p_0 到 p_1 的线段上。这里是一些运行示例:

```
Enter three points for p0, p1, and p2: 1 1 2.5 2.5 1.5 1.5 ↵ Enter
(1.5, 1.5) is on the line segment from (1.0, 1.0) to (2.5, 2.5) ↵ Enter
```

```
Enter three points for p0, p1, and p2: 1 1 2 2 3.5 3.5 ↵ Enter
(3.5, 3.5) is not on the line segment from (1.0, 1.0) to (2.0, 2.0)
```