

```

23             Integer.parseInt(args[2]));
24         break;
25     case '/': result = Integer.parseInt(args[0]) /
26                 Integer.parseInt(args[2]);
27 }
28
29 // Display result
30 System.out.println(args[0] + ' ' + args[1] + ' ' + args[2]
31                     + " = " + result);
32 }
33 }

```

`Integer.parseInt(args[0])` (第 16 行) 将一个数字字符串转换为一个整数。该字符串必须由数字构成, 否则, 程序会非正常中断。

我们使用 `.` 符号用于乘法, 而不是通常的 `*` 符号。原因是当符号 `*` 用于命令行时表示当前目录下的所有文件。在使用命令 `java Test *` 之后, 下面的程序就会显示当前目录下的所有文件:

```

public class Test {
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++)
            System.out.println(args[i]);
    }
}

```

为了解决这个问题, 我们需要使用其他符号来用于乘法操作。

☛ 复习题

7.29 本书声明 `main` 方法为:

```
public static void main(String[] args)
```

它可以替换为下面行中的哪些呢?

```

public static void main(String args[])
public static void main(String[] x)
public static void main(String x[])
static void main(String x[])

```

7.30 给出使用下面命令调用时, 以下程序的输出。

1. `java Test I have a dream`
2. `java Test "1 2 3"`
3. `java Test`

```

public class Test {
    public static void main(String[] args) {
        System.out.println("Number of strings is " + args.length);
        for (int i = 0; i < args.length; i++)
            System.out.println(args[i]);
    }
}

```

关键术语

anonymous array (匿名数组)

array (数组)

array initializer (数组初始化语法)

binary search (二分查找)

garbage collection (垃圾回收)

index (下标)

indexed variable (下标变量)

linear search (线性查找)

off-by-one error (过一错误)

selection sort (选择排序)

本章小结

1. 使用语法 `elementType[] arrayRefVar` (元素类型[]数组引用变量) 或 `elementType arrayRefVar[]` (元素类型 数组引用变量[]) 声明一个数组类型的变量。尽管 `elementType arrayRefVar[]` 也是合法的, 但还是推荐使用 `elementType[] arrayRefVar` 风格。
2. 不同于基本数据类型变量的声明, 声明数组变量并不会给数组分配任何空间。数组变量不是基本数据类型变量。数组变量包含的是对数组的引用。
3. 只有创建数组后才能给数组元素赋值。可以使用 `new` 操作符创建数组, 语法如下: `new elementType[arraySize]` (数据类型 [数组大小])。
4. 数组中的每个元素都是使用语法 `arrayRefVar[index]` (数组引用变量 [下标]) 表示的。下标必须是一个整数或一个整数表达式。
5. 创建数组之后, 它的大小就不能改变, 可以使用 `arrayRefVar.length` 得到数组的大小。由于数组的下标总是从 0 开始, 所以, 最后一个下标总是 `arrayRefVar.length-1`。如果试图引用数组界外的元素, 就会发生越界错误。
6. 程序员经常会错误地用下标 1 访问数组的第一个元素, 但是, 实际上这个元素的下标应该是 0。这个错误称为下标过 1 错误 (index off-by-one error)。
7. 当创建一个数组时, 如果其中的元素的基本数据类型是数值型, 那么赋默认值 0。字符类型的默认值为 `'\u0000'`, 布尔类型的默认值为 `false`。
8. Java 有一个称为数组初始化语法 (array initializer) 的简捷表达式, 它将数组的声明、创建和初始化合并为一条语句, 其语法为:
元素类型 [] 数组引用变量 = {value0,value1,...,valuek}
9. 将数组参数传递给方法时, 实际上传递的是数组的引用; 更准确地说, 被调用的方法可以修改调用者的原始数组的元素。
10. 如果数组是排好序的, 对于查找数组中的一个元素而言, 二分查找比线性查找更加高效。
11. 选择排序找到列表中最小的数字, 并将其和第一个数字交换。然后在剩下的数字中找到最小的, 和剩下列表的第一个元素交换, 继续这个步骤, 直到列表中只剩下一个数字。

测试题

在线回答本章节的测试题, 位于 www.cs.armstrong.edu/liang/intro10e/quiz.html。

编程练习题

7.2 ~ 7.5 节

*7.1 (指定等级) 编写一个程序, 读入学生成绩, 获取最高分 `best`, 然后根据下面的规则赋等级值:

- 如果分数 $\geq \text{best}-10$, 等级为 A
- 如果分数 $\geq \text{best}-20$, 等级为 B
- 如果分数 $\geq \text{best}-30$, 等级为 C
- 如果分数 $\geq \text{best}-40$, 等级为 D
- 其他情况下, 等级为 F

程序提示用户输入学生总数，然后提示用户输入所有的分数，最后显示等级得出结论。下面是一个运行示例：

```
Enter the number of students: 4 Enter
Enter 4 scores: 40 55 70 58 Enter
Student 0 score is 40 and grade is C
Student 1 score is 55 and grade is B
Student 2 score is 70 and grade is A
Student 3 score is 58 and grade is B
```

7.2 (倒置输入的数) 编写程序，读取 10 个整数，然后按照和读入顺序相反的顺序将它们显示出来。

**7.3 (计算数字的出现次数) 编写程序，读取在 1 到 100 之间的整数，然后计算每个数出现的次数。假定输入是以 0 结束的。下面是这个程序的一个运行示例：

```
Enter the integers between 1 and 100: 2 5 6 5 4 3 23 43 2 0 Enter
2 occurs 2 times
3 occurs 1 time
4 occurs 1 time
5 occurs 2 times
6 occurs 1 time
23 occurs 1 time
43 occurs 1 time
```

🔑 注意：如果一个数出现的次数大于一次，就在输出时使用复数“times”。

7.4 (分析成绩) 编写一个程序，读入个数不确定的考试分数，并且判断有多少个分数是大于或等于平均分，多少个分数是低于平均分的。输入一个负数表示输入的结束。假设最高分为 100。

**7.5 (打印不同的数) 编写一个程序，读入 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，读入一个数，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
```

*7.6 (修改程序清单 5-15) 程序清单 5-15 通过检验 2, 3, 4, 5, 6, ..., $n/2$ 是否是数 n 的因子来判断 n 是否是素数。如果找到一个因子， n 就不是素数。判断 n 是否素数的另一个更有效的方法是：检验小于等于 \sqrt{n} 的素数是否都能整除 n 。如果不能，则 n 就是素数。使用这个方法改写程序清单 5-15 以显示前 50 个素数。需要使用一个数组存储这些素数，然后再检查它们是否是 n 的可能的因子。

*7.7 (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。

7.6 ~ 7.8 节

7.8 (求数组的平均值) 编写两个重载的方法，使用下面的方法头返回一个数组的平均数：

```
public static int average(int[] array)
public static double average(double[] array)
```

编写测试程序，提示用户输入 10 个 `double` 型值，调用这个方法，然后显示平均值。

7.9 (找出最小元素) 编写一个方法，使用下面的方法头求出一个整数数组中的最小元素：

```
public static double min(double[] array)
```

编写测试程序，提示用户输入十个数字，调用这个方法返回最小值，显示其最小值。下面是

该程序的运行示例：

```
Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2 Enter
The minimum number is: 1.5
```

- 7.10 (找出最小元素的下标) 编写一个方法，求出整数数组中最小元素的下标。如果这样的元素个数大于1，则返回最小的下标。使用下面的方法头：

```
public static int indexOfSmallestElement(double[] array)
```

编写测试程序，提示用户输入10个数字，调用这个方法，返回最小元素的下标，然后显示这个下标值。

- *7.11 (统计学方面：计算标准差) 编程练习题 5.45 计算数字的标准差。本题使用一个和它不同但等价的公式来计算 n 个数的标准差。

$$\text{平均值} = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n} \quad \text{标准差} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{mean})^2}{n-1}}$$

要用这个公式计算标准差，必须使用一个数组存储每一个数，因此，可以在获取平均值后使用它们。

程序应该包含下面的方法：

```
/** Compute the deviation of double values */
public static double deviation(double[] x)

/** Compute the mean of an array of double values */
public static double mean(double[] x)
```

编写测试程序，提示用户输入10个数字，然后显示平均值和标准差，如下面的运行示例所示：

```
Enter ten numbers: 1.9 2.5 3.7 2 1 6 3 4 5 2 Enter
The mean is 3.11
The standard deviation is 1.55738
```

- *7.12 (倒置数组) 7.7 节中的 `reverse` 方法通过把数组复制到新数组中实现数组的倒置。改写方法将传递到实参的数组倒置，然后返回这个数组。编写一个测试程序，提示用户输入十个数字，调用这个方法倒置这些数字，然后显示这些数字。

7.9 节

- *7.13 (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)
```

- 7.14 (计算 gcd) 编写一个方法，返回个数不确定的整数的最大公约数。指定这个方法头如下所示：

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

编写测试程序，提示用户输入5个数字，调用该方法找出这些数的最大公约数，并显示这个最大公约数。

7.10 ~ 7.12 节

- 7.15 (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
```

7.16 (执行时间) 编写程序, 随机产生 100 000 个整数值和一个关键字。估算一下调用程序清单 7-6 中的 `linearSearch` 方法的执行时间。对该数组进行排序, 然后估算调用程序清单 7-7 中的 `binarySearch` 方法的执行时间。可以使用下面的代码模板获取执行时间:

```
long startTime = System.currentTimeMillis();
perform the task;
long endTime = System.currentTimeMillis();
long executionTime = endTime - startTime;
```

**7.17 (对学生排序) 编写一个程序, 提示用户输入学生个数、学生姓名和他们的成绩, 然后按照学生成绩的降序打印学生的姓名。

**7.18 (冒泡排序) 使用冒泡排序算法编写一个排序方法。冒泡排序算法遍历数组几次。在每次遍历中, 对相邻的两个元素进行比较。如果这一对元素是降序, 则交换它们的值; 否则, 保持值不变。由于较小的值像气泡一样逐渐“浮向”顶部, 同时较大的值“沉向”底部, 所以, 这种技术称为冒泡排序法 (bubble sort) 或下沉排序法 (sinking sort)。编写一个测试程序, 读取 10 个 `double` 型的值, 调用这个方法, 然后显示排好序的数字。

**7.19 (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
```

*7.20 (修改选择排序法) 在 7.11 节中, 使用的是选择排序法对数组排序。选择排序法重复地在当前数组中找到最小值, 然后将这个最小值与该数组中的第一个数进行交换。改写这个程序, 重复地在当前数组中找到最大值, 然后将这个最大值与该数组中的最后一个数进行交换。编写测试程序, 读取 10 个 `double` 型的数字, 调用该方法, 然后显示排好序的数字。

***7.21 (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。它是一个三角形形状的均匀放置钉子 (或钩子) 的直立板子, 如图 7-13 所示。

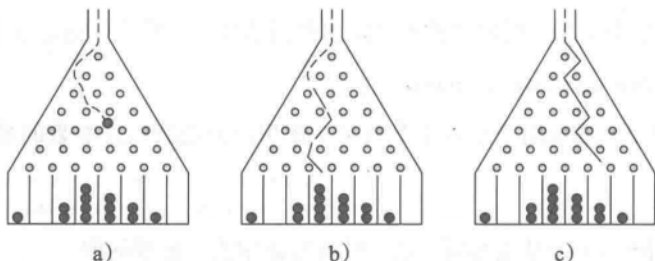


图 7-13 每个球都选取一个随机路径, 然后掉入一个槽中

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 7-13b 中球的路径是 LLRRLLR，而在图 7-13c 中球的路径是 RLRRLLR。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 the number of slots in the bean machine: 8

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。

*****7.22 (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。一个示例输出如右图所示。

Q							
			Q				
				Q			Q
		Q					
	Q					Q	
				Q			
			Q				

*****7.23 (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 S5 从 L5 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。

*****7.24 (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
```

7.25 (代数问题：解二次方程式) 使用下面的方法头编写一个解二次方程式的方法：

```
public static int solveQuadratic(double[] eqn, double[] roots)
```

二次方程式 $ax^2+bx+c=0$ 的系数都传给数组 eqn，然后将两个非复数的根存在 roots 里。方法返回根的个数。参见编程练习题 3.1 了解如何解二次方程。

编写程序，提示用户输入 a 、 b 和 c 的值，然后显示实数根的个数以及所有的实数根。

- 7.26 (完全相同的数组) 如果两个数组 `list1` 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 `list1` 和 `list2` 是完全相同的。使用下面的方法头编写一个方法，如果 `list1` 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 list2: 5 2 5 6 1 6
Two lists are strictly identical
```

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

- 7.27 (相同的数组) 如果两个数组 `list1` 和 `list2` 的内容相同，那么就说它们是相同的。使用下面的方法头编写一个方法，如果 `list1` 和 `list2` 是相同的，该方法就返回 `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，然后显示它们两个是否相同。下面是运行示例。注意，输入的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。

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

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

- *7.28 (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。
- *7.29 (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。
- *7.30 (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入输入的大小，即列表中值的个数。这里是一个运行示例。

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

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


****7.31** (合并两个有序列表) 编写下面的方法, 将两个有序列表合并成一个新的有序列表。

```
public static int[] merge(int[] list1, int[] list2)
```

只进行 `list1.length+list2.length` 次比较来实现该方法。编写一个测试程序, 提示用户输入两个有序列表, 然后显示合并的列表。下面是一个运行示例。注意, 输入的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。

```
Enter list1: 5 1 5 16 61 111
Enter list2: 4 2 4 5 6
The merged list is 1 2 4 5 5 6 16 61 111
```

****7.32** (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。

```
public static int partition(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 `[5,2,9,3,8]`, 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。

```
Enter list: 8 10 1 5 16 61 9 11 1
After the partition, the list is 9 1 5 1 10 61 11 16
```

***7.33** (文化: 中国生肖) 使用一个字符串数组存储动物名称, 来简化程序清单 3-9 的程序。

****7.34** (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。

*****7.35** (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
(Guess) Enter a letter in word p***** > r
(Guess) Enter a letter in word pr***r** > p
p is already in the word
(Guess) Enter a letter in word pr***r** > o
(Guess) Enter a letter in word pro*r** > g
(Guess) Enter a letter in word progr** > n
n is not in the word
(Guess) Enter a letter in word progr** > m
(Guess) Enter a letter in word progr*m > a
The word is program. You missed 1 time
Do you want to guess another word? Enter y or n>
```