

```

23      // Display the prime number and increase the count
24      if (isPrime) {
25          count++; // Increase the count
26
27          if (count % NUMBER_OF_PRIMES_PER_LINE == 0) {
28              // Display the number and advance to the new line
29              System.out.println(number);
30          }
31          else
32              System.out.print(number + " ");
33      }
34
35      // Check if the next number is prime
36      number++;
37  }
38  }
39  }

```

The first 50 prime numbers are

2 3 5 7 11 13 17 19 23 29

31 37 41 43 47 53 59 61 67 71

73 79 83 89 97 101 103 107 109 113

127 131 137 139 149 151 157 163 167 173

179 181 191 193 197 199 211 223 227 229

对编程新手而言这是一个复杂的例子。开发编程解决方案来解决这个问题或其他很多问题的关键之处在于要把问题分解成子问题，然后逐个地开发出每个子问题的解决方案。不要一开始就试图开发出一个完整的解决方案。而是应该首先编写代码判断一个给定的数是否是素数，然后扩展这个程序，再在循环中判断其他数是否是素数。

为了判断一个数是否是素数，检验该数是否能被 2 到 $\text{number}/2$ 之间并包括 2 和 $\text{number}/2$ 的整数整除（第 16 ~ 21 行）。如果能被整除，那它就不是素数（第 18 行）；否则，它就是一个素数。若是素数，就显示该数。若 count 能被 10 整除（第 27 ~ 30 行），就转入一个新行。当计数器 count 达到 50 时，程序终止。

程序在第 19 行使用 `break` 语句，一旦发现 number 不是素数，就立即退出 `for` 循环。也可以不用 `break` 语句，改写这个循环（第 16 ~ 21 行），如下所示：

```

for (int divisor = 2; divisor <= number / 2 && isPrime;
    divisor++) {
    // If true, the number is not prime
    if (number % divisor == 0) {
        // Set isPrime to false, if the number is not prime
        isPrime = false;
    }
}

```

然而，在本例中，使用 `break` 语句可以使程序更简单、更易读。

关键术语

`break` statement (`break` 语句)

`continue` statement (`continue` 语句)

`do-while` loop (`do-while` 循环)

`for` loop (`for` 循环)

`infinite` loop (无限循环、死循环)

input redirection (输入重定向)

iteration (迭代)

loop (循环)

loop body (循环体)

nested loop (嵌套循环)

off-by-one error (差一错误)

output redirection (输出重定向)

posttest loop (后测循环)

pretest loop (前测循环)

sentinel value (标志值)

while loop (while 循环)


本章小结

1. 循环语句有三类：`while` 循环、`do-while` 循环和 `for` 循环。
2. 循环中包含重复执行的语句的部分称为循环体。
3. 循环体执行一次称为循环的一次迭代。
4. 无限循环是指循环语句被无限次执行。
5. 在设计循环时，既需要考虑循环控制结构，还需要考虑循环体。
6. `while` 循环首先检查循环继续条件。如果条件为 `true`，则执行循环体；如果条件为 `false`，则循环结束。
7. `do-while` 循环与 `while` 循环类似，只是 `do-while` 循环先执行循环体，然后再检查循环继续条件，以确定是继续还是终止。
8. `while` 和 `do-while` 循环常用于循环次数不确定的情况。
9. 标记值是一个特殊的值，用来标记循环的结束。
10. `for` 循环一般用在循环体执行次数固定的情况。
11. `for` 循环控制由三部分组成。第一部分是初始操作，通常用于初始化控制变量。第二部分是循环继续条件，决定是否执行循环体。第三部分是每次迭代后执行的操作，经常用于调整控制变量。通常，在控制结构中初始化和修改循环控制变量。
12. `while` 循环和 `for` 循环都称为前测循环 (`pretest loop`)，因为在循环体执行之前，要检测一下循环继续条件。
13. `do-while` 循环称为后测循环 (`posttest loop`)，因为在循环体执行之后，要检测一下这个条件。
14. 在循环中可以使用 `break` 和 `continue` 这两个关键字。
15. 关键字 `break` 立即终止包含 `break` 的最内层循环。
16. 关键字 `continue` 只是终止当前迭代。

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：**对每个问题都应该多读几遍，直到理解透彻为止。在编码之前，思考一下如何解决这个问题。然后将你的逻辑翻译成程序。

通常，一个问题可以有多种不同的解决方法。鼓励学生探索不同的解决方案。

5.2 ~ 5.7 节

- *5.1 (统计正数和负数的个数然后计算这些数的平均值) 编写程序，读入未指定个数的整数，判断读入的正数有多少个，读入的负数有多少个，然后计算这些输入值的总和及其平均值 (不对 0 计数)。当输入为 0 时，表明程序结束。将平均值以浮点数显示。下面是一个运行示例：

```
Enter an integer, the input ends if it is 0: 1 2 -1 3 0 Enter
The number of positives is 3
The number of negatives is 1
The total is 5.0
The average is 1.25
```

Enter an integer, the input ends if it is 0: 0

No numbers are entered except 0

5.2 (重复加法) 程序清单 5-4 产生了 5 个随机减法问题。改写该程序, 使它产生 10 个随机加法问题, 加数是两个 1 到 15 之间的整数。显示正确答案的个数和测验时间。

5.3 (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

5.4 (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

5.5 (千克与磅之间的互换) 编写一个程序, 并排显示下列两个表格。

千克	磅	磅	千克
1	2.2	20	9.09
3	6.6	25	11.36
...			
197	433.4	510	231.82
199	437.8	515	234.09

5.6 (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

**5.7 (财务应用程序: 计算将来的学费) 假设今年某大学的学费为 10 000 美元, 学费的年增长率为 5%。一年后, 学费将是 10 500 美元。编写程序, 计算 10 年后的学费, 以及从现在开始 10 年后算起, 4 年内总学费是多少?

5.8 (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。

*5.9 (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。

5.10 (找出能被 5 和 6 整除的数) 编写程序, 显示从 100 到 1000 之间所有能被 5 和 6 整除的数, 每行显示 10 个。数字之间用一个空格字符隔开。

5.11 (找出能被 5 或 6 整除, 但不能被两者同时整除的数) 编写程序, 显示从 100 到 200 之间所有能被 5 或 6 整除, 但不能被两者同时整除的数, 每行显示 10 个数。数字之间用一个空格字符隔开。

5.12 (求满足 $n^2 > 12\,000$ 的 n 的最小值) 使用 while 循环找出满足 n^2 大于 12 000 的最小整数 n 。

5.13 (求满足 $n^3 < 12\,000$ 的 n 的最大值) 用 while 循环找出满足 n^3 小于 12 000 的最大整数 n 。

5.8 ~ 5.10 节

*5.14 (计算最大公约数) 下面是求两个整数 n_1 和 n_2 的最大公约数的程序清单 5-9 的另一种解法: 首先找出 n_1 和 n_2 的最小值 d , 然后依次检验 $d, d-1, d-2, \dots, 2, 1$ 是否是 n_1 和 n_2 的公约数。

第一个满足条件的公约数就是 $n1$ 和 $n2$ 的最大公约数。编写程序，提示用户输入两个正整数，然后显示最大公约数。

- *5.15 (显示 ASCII 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 码表如附录 B 所示。数字之间用一个空格字符隔开。
- *5.16 (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。
- **5.17 (显示金字塔) 编写程序，提示用户输入一个在 1 到 15 之间的整数，然后显示一个金字塔形状的模式，如下面的运行示例所示：

```

Enter the number of lines: 7
      1
     2 1 2
    3 2 1 2 3
   4 3 2 1 2 3 4
  5 4 3 2 1 2 3 4 5
 6 5 4 3 2 1 2 3 4 5 6
7 6 5 4 3 2 1 2 3 4 5 6 7
  
```

- *5.18 (使用循环语句打印 4 个图案) 使用嵌套的循环语句，用四个独立的程序打印下面的图案：

图案 1	图案 2	图案 3	图案 4
1	1 2 3 4 5 6	1	1 2 3 4 5 6
1 2	1 2 3 4 5	2 1	1 2 3 4 5
1 2 3	1 2 3 4	3 2 1	1 2 3 4
1 2 3 4	1 2 3	4 3 2 1	1 2 3
1 2 3 4 5	1 2	5 4 3 2 1	1 2
1 2 3 4 5 6	1	6 5 4 3 2 1	1

- **5.19 (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
  
```

- *5.20 (打印 2 到 1000 之间的素数) 修改程序清单 5-15，打印 2 到 1000 之间、包括 2 和 1000 的所有素数，每行显示 8 个素数。数字之间用一个空格字符隔开。

综合题

- **5.21 (财务应用程序：比较不同利率下的贷款) 编写程序，让用户输入贷款总额和以年为单位的贷款期限，然后显示利率从 5% 到 8%，每次递增 1/8 的过程中，每月的支付额和总支付额。下面是一个运行示例：

```

Loan Amount: 10000
Number of Years: 5
Interest Rate    Monthly Payment    Total Payment
5.000%          188.71          11322.74
5.125%          189.29          11357.13
5.250%          189.86          11391.59
  
```

...		
7.875%	202.17	12129.97
8.000%	202.76	12165.84

计算月支付额的公式，请参见程序清单 2-9。

- **5.22** (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：


```


Loan Amount: 10000 
Number of Years: 1 
Annual Interest Rate: 7 

Monthly Payment: 865.26
Total Payment: 10383.21

Payment#    Interest    Principal    Balance
1            58.33       806.93       9193.07
2            53.62       811.64       8381.43
...
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);
}

```

- *5.23** (示例抵消错误) 当处理一个很大的数字以及一个很小的数字的时候，会产生一个抵消错误(cancellation error)。例如， $100\,000\,000.0 + 0.000\,000\,001$ 等于 $100\,000\,000.0$ 。为了避免抵消错误，从而获得更加精确的结果，谨慎选择计算的次序。比如，在计算下面的数列时，从右到左计算要比从左到右计算得到的结果更精确：

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

编写程序对上面的数列从左到右和从右到左计算的结果进行比较，这里取 $n=50000$ 。

- *5.24** (数列求和) 编写程序，计算下面数列的和：

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

- **5.25** (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 $i=10000, 20000, \dots, 100000$ 时 π 的值。

****5.26** (计算 e) 使用下面的数列可以近似计算 e ：

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

编写程序，显示当 $i=10000, 20000, \dots, 100000$ 时 e 的值。

 **提示：**由于 $i! = i \times (i-1) \times \dots \times 2 \times 1$ ，那么 $\frac{1}{i!} = \frac{1}{i(i-1)!}$ 。将 e 和通项 $item$ 初始化为 1，反复将新的

$item$ 加到 e 上。新的 $item$ 由前一个 $item$ 除以 i 得到，其中 $i=2, 3, 4, \dots$ 。

****5.27** (显示闰年) 编写程序，显示从 101 到 2100 期间所有的闰年，每行显示 10 个。数字之间用一个空格字符隔开，同时显示这期间闰年的数目。

****5.28** (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

****5.29** (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				

***5.30** (财务应用程序：复利值) 假设你每月在储蓄账户上存 100 美元，年利率是 5%。那么每月利率是 $0.05/12=0.00417$ 。在第一个月之后，账户上的值变成：

$$100 * (1 + 0.00417) = 100.417$$

第二个月之后，账户上的值变成：

$$(100 + 100.417) * (1 + 0.00417) = 201.252$$

第三个月之后，账户上的值变成：

$$(100 + 201.252) * (1 + 0.00417) = 302.507$$

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。

- *5.31 (财务应用程序:计算CD价值)假设你投资10 000美元投资一张CD,年获利率为5.75%。一个月后,这张CD价值为

$$10000 + 10000 * 5.75 / 1200 = 10047.92$$

两个月之后,这张CD价值为

$$10047.91 + 10047.91 * 5.75 / 1200 = 10096.06$$

三个月之后,这张CD价值为

$$10096.06 + 10096.06 * 5.75 / 1200 = 10144.44$$

依此类推。


编写程序,提示用户输入一个总数(例如:10 000)、年获利率(例如:5.75)以及月份数(例如:18),然后显示一个表格,如下面的运行示例所示:

```

Enter the initial deposit amount: 10000 Enter
Enter annual percentage yield: 5.75 Enter
Enter maturity period (number of months): 18 Enter

Month  CD Value
1      10047.92
2      10096.06
...
17     10846.57
18     10898.54
  
```

- **5.32 (游戏:彩票)修改程序清单3-8,产生一个两位整数的彩票。这个数中的两个整数是两个不同的数。

 提示:产生第一个数,使用循环不断产生第二个数,直到它和第一个数不同为止。

- **5.33 (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 $28=1+2+4+7+14$ 。10 000 以下的完全数有四个。编写程序,找出这四个完全数。

- ***5.34 (游戏:石头、剪刀、布)编程练习题3.17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。

- *5.35 (加法)编写程序,计算下面的和。

$$\frac{1}{1+\sqrt{2}} + \frac{1}{\sqrt{2}+\sqrt{3}} + \frac{1}{\sqrt{3}+\sqrt{4}} + \cdots + \frac{1}{\sqrt{624}+\sqrt{625}}$$

- **5.36 (商业应用程序:检测ISBN)使用循环简化编程练习题3.9。

- **5.37 (十进制到二进制)编写程序,提示用户输入一个十进制整数,然后显示对应的二进制值。在这个程序中不要使用Java的Integer.toString(int)方法。

- **5.38 (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Integer.toString(int)方法。

- *5.39 (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 5 000 美元的税率是 8%，下一个 5 000 美元是 10%，余下的是 12%。如果销售额是 25 000，提成则为 $5\,000 * 8\% + 5\,000 * 10\% + 15\,000 * 12\% = 2\,700$ 。

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。

5.40 (模拟：正面或反面) 编写程序，模拟抛硬币一百万次，显示出正面和反面的次数。

*5.41 (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，就把它赋值给 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
The largest number is 5
The occurrence count of the largest number is 4
```

*5.42 (财务应用程序：求出销售额) 如下改写编程练习题 5.39：

- 使用 for 循环替代 do-while 循环。
- 允许用户自己输入 COMMISSION_SOUGHT 而不是将它固定为一个常量。

*5.43 (数学方面：组合) 编写程序，显示从整数 1 到 7 中选择两个数字的所有组合，同时显示所有组合的总数。

```
1 2
1 3
...
...

The total number of all combinations is 21
```

*5.44 (计算机体系结构：比特级的操作) 一个 short 型值用 16 位比特存储。编写程序，提示用户输入一个短整型，然后显示这个整数的 16 比特形式。下面是一个运行示例：

```
Enter an integer: 5
The bits are 0000000000000101
```

```
Enter an integer: -5
The bits are 1111111111111011
```

🔑 **提示：**需要使用按位右移操作符 (>>) 以及按位 AND 操作符 (&)，详见附录 G。

**5.45 (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。

$$\text{平均值} = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

$$\text{方差} = \sqrt{\frac{\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{n}}{n-1}}$$

下面是一个运行示例：

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


- *5.46 (倒排一个字符串) 编写一个程序, 提示用户输入一个字符串, 然后以反序显示该字符串。

```
Enter a string: ABCD
The reversed string is DCBA
```

- *5.47 (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。下面是一个运行示例:

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

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

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

- *5.48 (处理字符串) 编写一个程序, 提示用户输入一个字符串, 显示奇数位置的字符。下面是一个运行示例:

```
Enter a string: Beijing Chicago
BiigCiao
```

- *5.49 (对元音和辅音进行计数) 假设字母 A、E、I、O、U 为元音。编写一个程序, 提示用户输入一个字符串, 然后显示字符串中元音和辅音的数目。

```
Enter a string: Programming is fun
The number of vowels is 5
The number of consonants is 11
```

- *5.50 (对大写字母计数) 编写一个程序, 提示用户输入一个字符串, 然后显示该字符串中大写字母的数目。

```
Enter a string: Welcome to Java
The number of uppercase letters is 2
```

- *5.51 (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 the second string: Welcome to programming
The common prefix is Welcome to
```

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