

`ult = index ? : -1;`

使用 Objective-C 编写预处理程序宏指令时，条件运算符非常便利。详细内容将在第 12 章“预处理程序”介绍。

5 练习

1. 编写一个程序，请求用户在终端输入两个整数。测试这两个值，确定第一个数是否可以被第二个数整除，然后在终端显示一条适当的消息。
2. 即使输入非法运算符或试图除以 0 时，代码清单 6-8A 也会显示累加器中的值。请修改该程序。
3. 修改 Fraction 类的 print 方法，使其同样可以显示整数（因此，分数 5/1 可以只显示成 5）。再修改这个方法，使其能将分子是 0 的分数显示成 0。
4. 编写一个程序，使其作为简单的打印计算器。该程序应该允许用户输入以下形式的表达式：

number operator

程序还应该能识别以下运算符：

*+ - * / S E*

其中，S 运算符通知程序将累加器设置成输入的数字模式，而 E 运算符通知程序终止运行。使用输入的数字作为第二个运算数对累加器的内容执行算术运算。以下是显示程序应该如何运算的示例操作：

Begin Calculations

10 S

= 10.000000

2 /

= 5.000000

55 -

= -50.000000

Set Accumulator to 10

Contents of Accumulator

Divide by 2

Contents of Accumulator

Subtract 55

```
100.25 S
= 100.250000
```

```
4 *
```

```
= 401.000000
```

```
0 E
```

```
= 401.000000
```

```
End of Calculations.
```

Set Accumulator to 100.25

Multiply by 4

End of program

确保这个程序可以检测除数为零的情况,还能对未知的运算符进行检查。使用代码清单 6-8 中开发的 `Calculator` 类来执行运算。注意:记住在使用 `scanf` 的格式化字符串中使用空格字符(如“%f %c”)来跳过输入中的空白字符。

5. 开发代码清单 5-9 用于翻转从终端输入数的各个位。然而,如果输入负数,这个程序就不能很好地运行。找出这种情况下发生了什么事情,然后修改这个程序,以便正确地处理负数。“正确地处理”指的是如果输入数-8645,程序的输出将是 5468-。
6. 编写一个程序,用于接受从终端输入的整数,提取并用英语显示这个数的每一个数字。

因此,如果用户输入 932,程序就会显示以下内容:

```
nine
three
two
```

- (记住,用户只输入一个 0 时,将显示 zero。)注意:这个练习很难!
7. 代码清单 6-10 存在几个低效的方面。其中一个低效的方面是由检查偶数引起的。因为任何大于 2 的偶数显然不能是素数,因此,可以简单地略过偶数作为可能的素数和可能的除数。内层的 `for` 循环也是低效的,因为始终使用 $2 \sim p-1$ 之间的所有 d 值除以 p 。如果在 `for` 循环的条件中添加用于判断 `isPrime` 值的测试,可以避免这种低效性。使用这种方式时,只要没有发现除数,而且 d 的值小于 p ,`for` 循环就将继续执行。修改代码清单 6-10,将这两种修改合并到一起,然后运行程序以验证它的运算。