

## 4.5 练习

1. 下列常量中，哪些是非法的？为什么？

123.456	0x10.5	0X0G1
0001	0xFFFF	123L
0xab05	0L	-597.25
123.5e2	.0001	+12
98.6F	98.7U	17777s
0996	-12E-12	07777
1234uL	1.2Fe-7	15,000
1.234L	197u	100U
0XABCDEFL	0xabcu	+123

2. 编写一个程序，使用以下公式将华氏温度 (F) 27° 转换成摄氏温度 (C):

$$C = (F - 32) / 1.8$$

不需要定义一个类来执行计算。只需要简单地列出表达式就满足要求。

3. 以下程序将输出什么结果？

```
#import <Foundation/Foundation.h>

int main (int argc, char *argv[])
{
    @autoreleasepool {
        char c, d;

        c = 'd';
        d = c;
        NSLog(@"d = %c", d);
    }
    return 0;
}
```

4. 编写一个程序，求以下多项式的值（计算表达式时，只需直接计算；因为在 Objective-C 中没有幂指操作符）：

$$3x^3 - 5x^2 + 6$$

(令  $x = 2.55$ )

5. 编写一个程序，求下列表达式的值，并显示其结果（记住要使用指数格式显示结果）。

$$(3.31 \times 10^{-8} + 2.01 \times 10^{-7}) / (7.16 \times 10^{-6} + 2.01 \times 10^{-8})$$

6. 复数包含两个部分：实部和虚部。如果  $a$  是实部， $b$  是虚部，那么符号

$a + bi$

可用来表示复数。

编写一个 Objective-C 程序，定义一个名为 Complex 的新类。依照为 Fraction 类创建的范例，为该类型定义以下方法：

```
-(void) setReal: (double) a;  
-(void) setImaginary: (double) b;  
-(void) print;      // 显示为 a + bi  
-(double) real;  
-(double) imaginary;
```

编写一个测试程序测试这个新类和各个方法。

7. 假设你正开发操作图形对象的函数库。从定义名为 Rectangle 的新类开始。目前，仅记录矩形的宽和高即可。开发一些方法用于设置矩形的宽和高、检索这些值以及计算矩形的面积和周长。假定这些矩形对象使用整数坐标栅格来描述矩形，例如，一台计算机屏幕。在这种情况下，假定矩形的宽和高都是整数值。

以下是 Rectangle 类的 @interface 部分：

```
@interface Rectangle: NSObject  
-(void) setWidth: (int) w;  
-(void) setHeight: (int) h;  
-(int) width;  
-(int) height;  
-(int) area;  
-(int) perimeter;  
@end
```

请编写 implementation 部分，并编写一个测试程序来测试新类的方法。

8. 修改代码清单 4-6 中的 add:、subtract:、multiply: 和 divide: 方法，使其返回累加器的结果值。测试这些新方法。
9. 完成练习 8 后，把以下方法添加到 Calculator 类中并测试它们：

```
-(double) changeSign; // 改变累加器的正负号  
-(double) reciprocal; // 累加器  
-(double) xSquared;   // 累加器的平方
```

10. 为代码清单 4-6 中的 Calculator 添加一项存储功能。实现以下方法声明并测试它们：

```
-(double) memoryClear;      // 清理内存  
-(double) memoryStore;     // 设置内存为累加器
```

```
-(double) memoryRecall;           // 设置累加器到内存  
-(double) memoryAdd: (double) value; // 添加值到内存  
-(double) memorySubtract: (double) value; // 与内存的值相减
```

为最后两组方法设置一个累加器,并能够对内存执行指定的操作。所有的方法都需要返回累加器的值。