第四讲
表达式

4.1算术运算符

- 表达式：由操作数和运算符组成的序列
- 算术运算符：+、-、*、/、%
- 优先级
  - 最好的方式：加括号
- 示例
  - 条形码验证
  - 校验码=9-(3*奇数位和+偶数位和-1)%10
  - 奇数位和: 0 + 3 + 0 + 1 + 1 + 3 = 8
  - 偶数位和: 1 + 8 + 0 + 5 + 7 = 21
  - 因此:　9-(3*8+21-1)%10=5

0 1 3 8 0 0 1 5 1 7 3
i1　　i10

4.2赋值运算符

- 简单赋值：=，将一个值存入到一个变量中

$v = e$

- e:常量、变量、表达式…

```
i = 5;              // i is now 5
j = i;              // j is now 5
k = 10 * i + j;     // k is now 55
```

- 如果v和e的类型不同，赋值时e的值会转换为v的类型

```
int i;
float f;

i = 72.99f;    // i is now 72
f = 136;       // f is now 136.0
```

4.2赋值运算符

- 复合赋值：基于变量原有值计算出新的值并重新赋值给这个变量

```
i += 2;    // same as i = i + 2;
```

- 其他复合赋值

+= -= *= /= %=

<<= >>= &= |=

- 注意

  - i += j 与 i =+ j
  - i * = j+k 与 i = i*j + k

4.3自增自减运算符

- ++：自增
- --：自减
- 前缀：++*i*
  - 使用之前使 *i* 的值增1
  - 马上将 *i* 加1，再用

```
i = 1;
printf("i is %d\n", ++i);    // prints "i is 2"
printf("i is %d\n", i);      // prints "i is 2"
```

- 后缀：*i*++
  - 使用之后使 *i* 的值增1
  - 先用旧值，之后再加1

```
i = 1;
printf("i is %d\n", i++);    // prints "i is 1"
printf("i is %d\n", i);      // prints "i is 2"
```

4.4 表达式求值

- 常用运算符

优先级	运算符	结合性
1	i++, i--	左
2	++i,--i,正号,负号	右
3	* / %	左
4	+ -	左
5	= *= /= %= += -=	右

举例

```
a = b += c++ - d + --e / -f
```

```
(a = (b += (((c++) - d) + ((--e) / (-f)))))
```