**程序设计实训 第二次作业**

徐浩博 2020010108

**编译运行**

开发环境：Windows 10, 64位操作系统

IDE：Microsoft Visual Studio 2019

编译运行方法：在IDE中直接编译运行

**算法分析与运行结果**

* **第1题**

在本题中，我考虑运算式一共有如下5个形式：

1. ((a○b)○c)○d

2. (a○b)○(c○d)

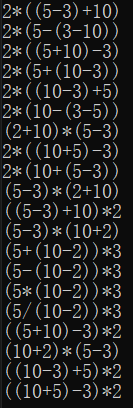
3. (a○(b○c))○d

4. a○((b○c)○d)

5. a○(b○(c○d))

其中a,b,c,d表示四个数，○表示加减乘除中的一个，因此我们可以利用枚举数字、运算符和运算式形式，计算每种枚举对应的运算结果来寻找满足运算结果为24的运算式。

首先枚举数字，我们采用dfs的枚举法；数字不能重复，则需要用一个bool数组在dfs时记录数字是否已被使用。由排列组合的知识：此种枚举共有A(4,4)=24种。

 其次枚举运算符，我们采用dfs的枚举法。由排列组合的知识：此种枚举共有43=64种。

最后枚举运算式形式，我们采用直接列举的方式。参照上面的内容，枚举有5种。

综上，以上的枚举共有24×64×5=7680种，因此运算的复杂度也不大。

对于给出的样例，程序运行结果如右图，可见程序运行结果正确。

* **第2题**

由于计算总是按输入n个数的顺序及添加的运算符自左向右进行的，因此唯一的变量只有数字之间的运算符。对于此题，只需枚举n个数之间的(n-1)个运算符，计算出每种枚举对应的计算式的值。如果计算的值等于D则问题解决，若计算的值大于D，则记录大于D的所有数的最小值。

考虑到本题，每一步运算结果都只与前一步运算结果有关，而与前序的运算过程无关，因此可以在dfs时记录前一步运算结果，本次运算只需在前一步运算结果上进行运算，不用在每次枚举结束从头运算，从而提高运算速度。

枚举共有2n-1种，考虑到n≤24，则枚举最多有223≈8×106种可能，考虑到算法的常数比较小，因此这种算法是可行的。

对于给出的样例，程序运行结果如下图，可见程序运行结果正确。

