24点判断程序设计

该设计的基本思路是将四个取值为[1~10]的数字a、b、c、d,组合为(ab)cd、(ab)(cd)、(abc)d、a(bc)d、ab(cd)、a(bcd)六种形式，再在中间相应的位置加入+-\*/符号，遍历完全，如果没有24这个结果，则该组合无法计算出24。以下是程序设计。

Create(\*L,a,b,c,d)

创建四个数字。

将a,b,c,d四个数字依次放入顺序表L中。

Group(\*L,\*L1,\*L2,\*L3,\*L4,\*L5,\*L6)

组合元素。

将L中的元素以六种形式插入括号并存入L1,L2,L3,L4,L5,L6中。

RPN(\*L)

翻译为逆波兰表达式。

将L中的中缀表达式翻译成逆波兰表达式。

Compute(\*L)

计算逆波兰表达式。

将逆波兰表达式计算出来，并返回结果。

Judge(\*L)

判断。

输入为经过Group()处理过的顺序表，利用依次遍历所有的符号插入并调用RPN()和Compute()进行运算，如果存在24，则返回1,否则返回0。

Main()

利用Create()生成取值范围为1～10的所有集合，并通过Group()和Judge()函数进行处理和判断，如果对于一个集合，结果出现了1,则判断可以生成24,否则，将输入的元素打印出来，表示无法生成24点。