

实验 2 栈的应用

问题 1：算术表达式求值演示

【问题描述】

表达式计算是实现程序设计语言的基本问题之一，也是栈的应用的一个典型例子。设计一个程序，演示用算符优先法对算术表达式求值的过程。

【基本要求】

以字符序列的形式从终端输入语法正确、不含变量的整数表达式。利用课件给出的算符优先关系，实现对算术四则混合运算表达式的求值，并演示在求值过程中操作符栈、操作数栈、输入字符和主要操作的变化过程。

输入：一个以“=”结尾的中缀算术表达式。运算符包括+、-、*、/、(、)、=，参加运算的数为正整数。例如， $3*(7-2)=$ 或者 $1024/(20+8)=$ 或者 $(20+2)*(6*(2+8))=$ 或者 $((2+4)*(5+7)*9+1)/2=$ 等等。

输出：表达式的计算结果。运算过程中操作符栈、操作数栈、输入字符和主要操作的内容。

【选作内容】

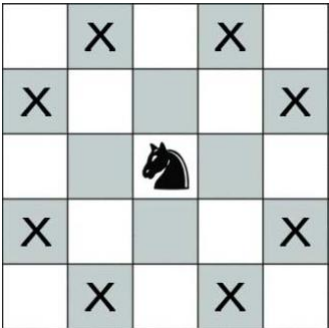
- (1) 扩充运算符集，如增加乘方、单目减、赋值等运算。
- (2) 操作数可以是变量。
- (3) 操作数可以是实数。
- (4) 图形化用户界面。

问题 2：骑士巡逻

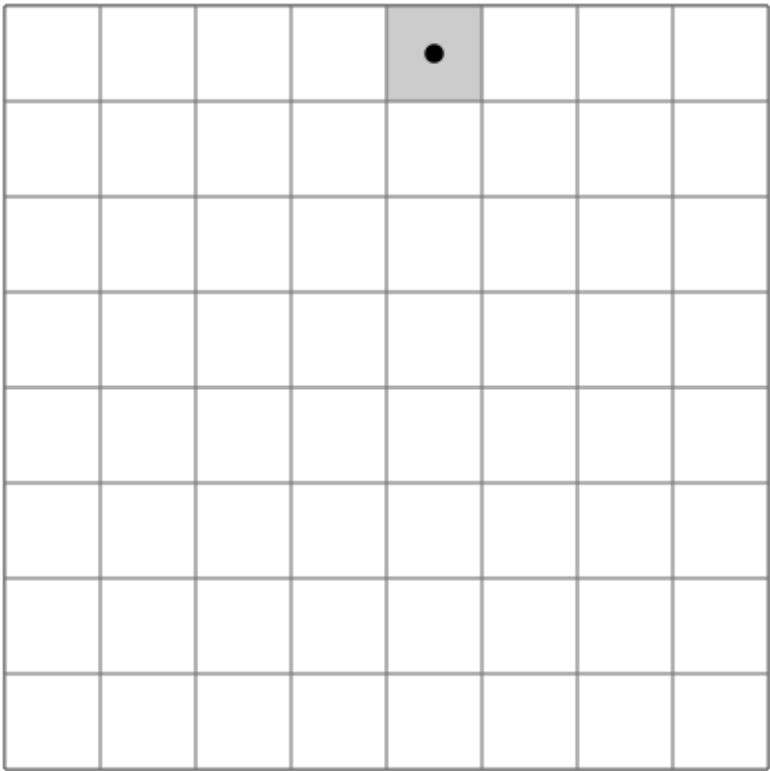
【问题描述】

骑士巡逻（Knight's tour）是指在按照国际象棋中骑士的规定走法走遍整个棋盘的每一个方格，而且每个网格只能够经过一次。假若骑士能够走回到最初位置，则称此巡逻为“封闭巡逻”，否则，称为“开巡逻”。

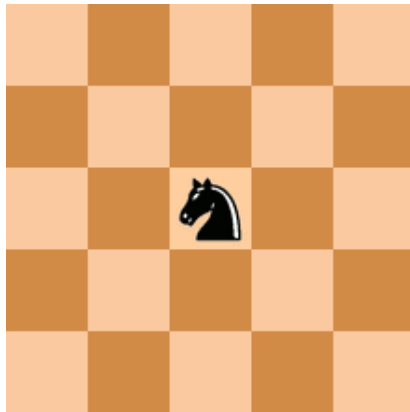
注：国际象棋中骑士的走法与中国象棋中马的走法相似，呈“日”字型或“L”字型。如下图所示。即每走一步，其行列坐标，一个变化 1，另一个变化 2。



一种开巡逻走法：



5×5 棋盘中的一种开巡逻走法：



【基本要求】

将骑士放在给定大小 ($n \times n$) 的国际象棋棋盘的给定位置上 (某一方格中), 骑士按照走棋规则进行移动, 要求每个方格只进入一次, 走遍棋盘上的所有方格。分别编写一个递归和一个非递归程序, 求出骑士的行走路线, 将数字 1, 2, 3, ..., $n \times n$ 依次填入这个棋盘上的所有方格, 数字表示路线上的第几步。

输入：棋盘的大小 n (只考虑正方形的棋盘, 即棋盘上共有 $n \times n$ 个方格); 骑士在棋盘上的起始位置 (x, y) 。

输出：骑士完成一次巡逻的路线 (即每个方格标注了从 1 到 $n \times n$ 不同数字的棋盘)。例如,

```
21  2   7  12  23
 8 13 22 17   6
 3 20  1  24 11
14  9 18  5 16
19  4 15 10 25
```

【选作内容】

- (1) 求出从一个起始点出发的多条巡逻路线。
- (2) 探讨每次选择位置的“最佳策略”, 以减少回溯的次数。
- (3) 演示寻找巡逻路线的回溯过程。