

生命游戏

生命游戏在一个无边界的矩形网格上进行，这个矩形网格中的每个单元可被占据，或者不被占据。被占据的单元称为活的，不被占据的单元称为死的。哪一个单元是活的是根据其周围活的邻居单元的数目而一代一代地发生变化的。一代一代转换的具体规则如下：

给定单元的邻居单元指的是与它在垂直、水平或对角方向上相接的 8 个单元。

如果一个单元是活的，则如果它具有 2 个或 3 个活的邻居单元，则此单元在下一代还是活的。

如果一个单元是活的，则如果它具有 0 个或 1 个、4 个或 4 个以上的活的邻居单元，则此单元在下一代会因为孤独或拥塞而死亡。

如果一个单元是死的，则如果它具有恰好有 3 个活的邻居，则此单元在下一代会复活，否则该单元在下一代仍然是死的。

(1) 要求编写程序，模拟任意一个初始输入配置以及代代更替的不同状态并进行显示。

(2) 修改以上程序,要求生成一个网格时，用“空格”和“x”分别表示网格中每一个死的单元和活的单元，并且可根据用户选择从键盘或者从文件读入初始配置。

附：大家还可到网上查阅运动员球衣分配、幻方的构造等问题，进行类似求解。

中缀表达式求解

中缀表达式是我们熟悉的表达式形式。为了能正确表示运算的先后顺序，中缀表达式中难免要出现括号。假设我们的表达式中只允许有圆括号。

读入一个浮点数为操作数的中缀表达式后，对该表达式进行运算。

要求中缀表达式以一个字符串的形式读入，可含有加、减、乘、除运算符和左、右括号，并假设该表达式以“#”作为输入结束符。

如输入“ $3.5*(20+4)-1\#$ ”，则程序运行结果应为 83。

要求可单步显示输入序列和栈的变化过程。并考虑算法的健壮性，当表达式错误时，要给出错误原因的提示。

附：大家可扩展考虑后缀表达式求值、中缀表达式求值、括号匹配等栈的综合应用

小猫钓鱼纸牌游戏

A 和 B 两个同学玩简单的纸牌游戏，每人手里有 n 张牌，两人轮流打牌并依次排列在桌面上，每次出掉手里的第 1 张牌，打牌后如果发现桌面上有跟刚才打出的牌的数字相同的牌，则把从相同的那张牌开始的全部牌按次序放在自己手里的牌的末尾。当一个人手中的牌先出完时，游戏结束，对方获胜。

如 n 为 5，A 手里的牌依次为 2 3 5 6 1，B 手里的牌依次为 1 5 4 2 9；

A 出 2；

B 出 1；

A 出 3；

B 出 5；

A 出 5，发现前面有一张 5，则把两个 5 都拿掉，这时他手里有 6 1 5 5；

桌子上的牌依次为 2 1 3；

B 出 4；

A 出 6；

B 出 2，发现前面有一张 2，则把从 2 开始的牌全部拿掉，这时他手里有 9 2 1 3 4 6 2；

桌子上没有牌了；

A 出 1；

B 出 9；

A 出 5；

B 出 2；

依次类推，直到某人先出完牌为止，则对方是胜者。

编写程序，利用栈和队列，判断谁是胜者。

长整数运算

- (1) 实现线性表顺序存储结构、链式存储结构的基本操作，包括顺序表、单链表、双链表、单循环链表、双循环链表等；
- (2) 设计并实现两个长整数的加、减、乘运算。

查找算法的实现及性能测试与比较

在顺序线性表中存放 n 个整数， n 的值由用户输入确定，线性表可以是有序表或无序表。

比较各查找算法在不同情况下的时间性能。

各查找算法的实测时间性能包括两个指标：算法执行的绝对时间和关键字的平均比较次数。

各查找算法要求评测查找成功与不成功的两种情形。

为了能比较出各种查找算法执行的绝对时间，需要对表中的数据进行较大量的查找，设为 m 次， m 的值也由用户输入确定。当输入 m 为 1000000 时，则对线性表作 1000000 次查找。

(1) 比较在有序表和无序表中进行顺序查找时，查找成功和查找失败时的算法执行的绝对时间和关键字的平均比较次数。

(2) 比较在同一有序表中进行顺序查找和二分查找时的时间性能。

(3) 比较在同一有序表中进行非递归二分查找和递归二分查找的时间性能。

排序算法的实现及性能测试及比较

[问题描述]在书中，各种内部排序算法的时间复杂度分析结果只给出了算法执行时间的阶，或大概执行时间。试通过具体数据比较各种算法的关键字比较次数和记录移动次数，以取得直观感受。

要求：

(1) 编写程序创建一些整数文件用于排序。创建的这些文件可以根据需要生成不同的长度，如长度分别为 20，200 和 2000，以正序、逆序、随机顺序的方式创建这些文件，通过把所有这些测试数据保存在文件中（而不是每次在测试程序时用随机数生成），可以使用同样的数据去测试不同的方法，因此会更易于比较这些方法的性能。

(2) 数据表采用顺序存储结构，实现插入排序，选择排序，希尔排序，归并排序，快速排序，堆排序等排序，并对这些算法的实现效率进行比较和分析。

(3) 排序的指标包含：运行时间，比较次数，移动次数。

二叉树

产生一个菜单驱动的演示程序，用以说明二叉树的使用。元素由单个键组成，键为单个字符。用户能演示的二叉树基本操作至少包括：构造二叉树，按先序、中序、后序、层序遍历这棵二叉树，求二叉树的深度、宽度，统计度为 0，1，2 的结点数等。

二叉树采用链式存储结构。

对二叉查找树做上述工作，且增加以下操作：插入、删除给定键的元素、查找目标键。

迷宫求解

一般的迷宫可表示为一个二维平面图形，将迷宫的左上角作为入口，右下角作为出口。迷宫问题求解的目标是寻找一条从入口点到出口点的通路。

例如，可设计一个 8×8 矩阵 `maze[8][8]` 来表示迷宫，如下所示

```
0  1  0  0  0  0  1  1
0  0  0  1  0  0  1  0
1  0  1  0  1  0  1  1
1  0  1  0  1  1  0  1
0  1  1  1  1  1  1  0
1  0  0  1  1  0  0  0
1  0  1  0  0  0  1  1
1  0  1  1  0  1  0  0
```

左上角 `maze[0][0]` 为起点，右下角 `maze[7][7]` 为终点；设“0”为通路，“1”为墙。假设一只老鼠从起点出发，目的为右下角的终点，可向“上、下、左、右、左上、左下、右上、右下”8个方向行走。

设计一个程序，能自动生成或手动生成一个 8×8 矩阵，针对这个矩阵，程序判断能否从起点经过迷宫走到终点。如果能，请给出一种走出迷宫的路径。

图-社交网络模型

设计并实现一个社交网络模型图。

要求：

- (1) 每个人的信息是一个顶点，两个人相互认识则构成边。
- (2) 根据输入的任意两个人的信息，给出他们之间的联系路径，最少经过多少人构成联系。
- (3) 可根据自己的创意添加更多的功能。

图-拯救 007

看过 007 系列电影的人们很熟悉 James Bond 这个世界上最著名的特工了。在电影 “Live and Let Die” 中 James Bond 被一组毒品贩子捉住并且关到湖中心的一个小岛上，而湖中有很多凶猛的鳄鱼。这是 James Bond 做出来最惊心动魄的事情来逃脱——他跳到了最近的鳄鱼的头上，在鳄鱼还未反应过来，他又跳到另一只鳄鱼的头上……最后他终于安全地跳到了湖岸上。

假设湖是 100×100 的正方形，设湖中心在 $(0, 0)$ ，湖的东北角坐标是 $(50, 50)$ 。湖中心的圆形小岛的圆心在 $(0, 0)$ ，直径是 15。一些凶残的鳄鱼分布在湖中不同的位置。现已知湖中鳄鱼的位置（坐标）和 James Bond 可以跳的最大距离，请你告诉 James Bond 一条最短的到达湖边的路径。他逃出去的路径长度等于他跳的次数。

