# 第三次作业

题目内容：

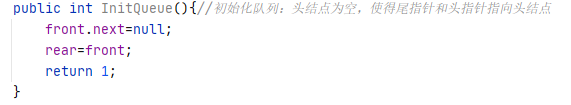
1. 我们每个人去银行办业务，要做的第一件事是在取号机上取号等待服务。其实，银行取号机是一个简单的队列，取号过程就是进队过程；而接受服务窗口服务过程就是出队过程。请同学们用单链表存储和管理方式实现这个服务功能。其中，主要功能包括队列初始化、入队操作和出队操作。
2. 栈操作也是我们日常生活中常见的例子，尤其在操作系统的CPU调度和存储管理等中应用非常普遍。所以，它也是我们同学在今后设计中最常用抽象数据类型之一。本例中，请同学们用栈操作来实现一组有序数据的倒排输出问题。例如，已有有序数据为“A、B、C、1、2、3、4、5、6、7”，我们需要用栈操作的先进后出的特性来实现它们的倒排输出，即“7、6、5、4、3、2、1、C、B、A”。
3. 请用栈操作算法实现探索迷宫过程。
4. 认真对比分析线性表、队列和栈这三个线性关系抽象数据类型的异同；再对比它们在设计和操作过程中的异同；最后，试试看，第一个实验项目中，哪些地方可以用队列或栈来代替其线性表？替换后，在功能实现和效率方面会带来哪些变化？！

第一题：

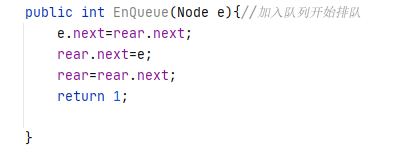
分析：题设内容要求我们初始化队列来解决银行取号问题，我设计的队列包括，front队首指针，rear队尾指针，在front处插入结点，在rear处删除结点，结点设计为一个包含取得的号码与是否已经获得了服务的数据类型，其中还有指向下一个结点的next指针。

操作：

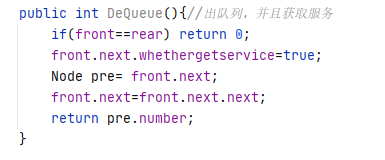
初始化队列



如注释



使得新结点指向原队尾结点的后继（也就是null），原队尾结点的后继指向新结点，然后队尾指针向后移。



首先判断，如果队尾和队首指针重合，返回0提示报错，将要出队列的结点设定为已获得服务，同时预存要出队列的结点，将队首指针连接到新的队列的第一个结点上，然后返回要出队列结点的号。

第二题

分析：

想要将一串字符倒置，最方便的是引用栈的先进后出的特性，先将字符挨个压栈，然后再依次出栈，出栈的同时输出字符，就能达到字符串倒转的功能。设计栈顶指针rear用来压栈出栈，设计栈底指针用来充当头结点

操作：



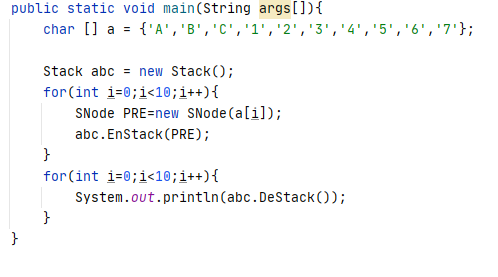
入栈：

让栈顶指针的后继指向新结点，栈顶指针后移

出栈

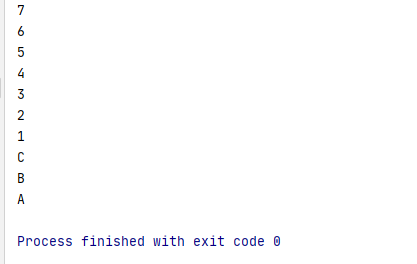
设置辅助指针一直寻找到栈顶指针之前，将栈顶指针的字符存储下来，然后top前移，返回应该出栈的字符

main函数



用字符数组存储字符，然后依次进栈，再依次出栈，实现倒转

实验结果截图



题目三

分析：题目说要实现迷宫算法，根据书本的知识和我个人的调整，我使用二维int数组表示迷宫，每处方块通过不同数字表达状态，如0是无障碍且未经过，1是有障碍，2是已经经过了，3是从这个方块退回过，然后设计一个栈用来存储每一步的信息，用数据结构包括一个表示坐标的数组，和表示移动方向的数字（用1，2，3，4分别表示东西南北），每一步都包含了这一步是从哪一个方块走的，和这一步是向哪个方向走的两部分信息，而且使用数字表示方向的一个好处是，我可以保证已经尝试过的方向不会再走第二次，得出的答案会尽可能地少退回，按照先尝试东，再尝试南，再尝试西，再尝试北，如果四个方向都不行的话再进行退回，继续尝试上一步还未尝试过地方向。而建立的栈，则是为了在退回的时候方便取出上一步的信息。

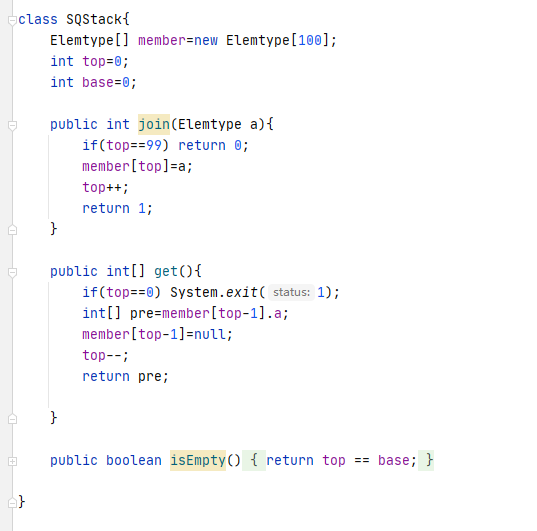
核心函数

移动



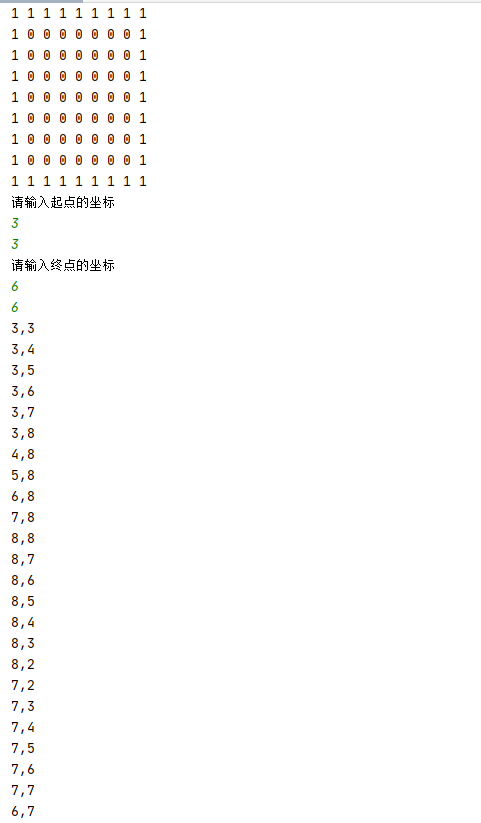
如注释所说，尝试四种方向移动，如果不行的话，则退回上一步

栈相关



该题运用的是顺序栈，用一个Elemtype数组保存信息，大体思路与书上类似。主要作用还是为了达成元素先进后出这一特性，为在迷宫中回退提供良好的数据结构。

实验结果示例



结果是从起点的坐标依次输入到终点的坐标，这构成一条出迷宫的路径

题目四；

队列和栈是特殊的线性表，队列和栈能做到的操作，线性表都能做到，但是线性表不管是链表还是顺序表，与队列和栈相比在操作的效率上都显得过低了，而且对于操作来说过于复杂，调用者需要考虑的事过多，在现实生活中的很多操作实际上不需要做到在线性表内的每一处都进行插入或者删除，只要能将先进先出和先进后出的两个需求实现，我们就不需要多余地去采用普通的线性表。

|  |  |  |  |
| --- | --- | --- | --- |
| 链式结构 | 线性表 | 队列 | 栈 |
| 插入 | 用辅助指针反复后移，在插入处进行指针操作 | 队尾指针操作即可 | 栈顶指针操作即可 |
| 删除 | 用辅助指针反复后移，在删除处进行指针操作 | 队头指针操作即可 | 辅助指针后移直至到达栈顶前进行指针操作 |
| 查询 | 指针反复后移 | 一般不需要查询，出队即为查询 | 一般不需要查询，出栈即为查询 |

|  |  |  |  |
| --- | --- | --- | --- |
| 顺序结构 | 线性表 | 队列 | 栈 |
| 插入 | 插入点后所有元素后移，再对插入点赋值 | 直接操作即可，但是有队列无法插入而数组实际未满地可能 | 直接操作即可，只需要更改栈顶 |
| 删除 | 删除点后所有元素前移 | 队头后移即可 | 直接操作即可，只需要更改栈顶 |
| 查询 | 直接查询 | 一般不需要查询，出队即为查询 | 一般不需要查询，出栈即为查询 |