

数据结构实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 题目 | 实验二：栈和队列的基本操作及其应用 | | |
|  |  | | |
|  |  |  |  |
| 学院 |  | 年级 |  |
| 专业 |  | 学号 |  |
| 姓名 |  | | |
| 指导老师 | 刘坤 | | |

2025年 月 日

实验二 栈和队列的基本操作及其应用

**一、实验目的**

1、掌握栈和队列的顺序存储结构和链式存储结构，以便在实际中灵活应用。

2、掌握栈和队列的特点，即后进先出和先进先出的原则。

3、掌握栈和队列的基本运算，如：入栈与出栈，入队与出队等运算在顺序存储结构和链式存储结构上的实现。

**二、实验内容**

本次实验提供4个题目，每个题目都标有难度系数，\*越多难度越大，完成题目一和题目二，题目三和题目四学生可以根据自己的情况任选一个！

题目一：回文判断（\*）

[问题描述]

对于一个从键盘输入的字符串，判断其是否为回文。回文即正反序相同。如“abba”是回文，而“abab”不是回文。

[基本要求]

（1）数据从键盘读入；

（2）输出要判断的字符串；

（3）利用栈的基本操作对给定的字符串判断其是否是回文，若是则输出“Yes”，否则输出“No”。

[测试数据]

由学生任意指定。

题目二：顺序栈和循环队列基本操作（\*）

[基本要求]

1. 实现栈的基本操作

六项基本操作的机制是：初始化栈：init\_stack(S)；判断栈空：stack\_empty(S)；取栈顶元素：stack\_top(S,x)；入栈：push\_stack(S,x)；出栈：pop\_stack(S)；判断栈满：stack\_full(S)

2、实现队列的基本操作

六项基本操作的机制是：初始化队列：init\_queue(Q)；判断队列是否为空：queue\_empty(Q)；取队头元素：queue\_front(Q,x)；入队：enqueue(Q,x)；出队：outqueue(Q,x)；判断队列是否为满：queue\_full(Q)

[测试数据]

由学生任意指定。

题目三：商品货架管理（\*\*）

[问题描述]

商店货架以栈的方式摆放商品。生产日期越近的越靠近栈底，出货时从栈顶取货。一天营业结束，如果货架不满，则需上货。入货直接将商品摆放到货架上，则会使生产日期越近的商品越靠近栈顶。这样就需要倒货架，使生产日期越近的越靠近栈底。  
[基本要求]

设计一个算法，保证每一次上货后始终保持生产日期越近的商品越靠近栈底。

[实现提示]

可以用一个队列和一个临时栈作为周转。

[测试数据]

由学生任意指定。

**三、实验前的准备工作**

1、掌握栈的逻辑结构和存储结构。

2、熟练掌握栈的出栈、入栈等操作。

3、掌握队列的逻辑结构和存储结构。

4、熟练掌握队列的出队、入队等操作

**四、实验报告要求**

1、实验报告要按照实验报告格式规范书写。

\*2、写出算法设计思路。

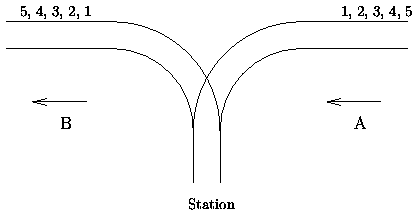
3、实验上要写出多批测试数据的运行结果。

4、结合运行结果，对程序进行分析。

题目四：**Rails（ACM训练题）**（\*\*\*）

**Description**

There is a famous railway station in PopPush City. Country there is incredibly hilly. The station was built in last century. Unfortunately, funds were extremely limited that time. It was possible to establish only a surface track. Moreover, it turned out that the station could be only a dead-end one (see picture) and due to lack of available space it could have only one track.



The local tradition is that every train arriving from the direction A continues in the direction B with coaches reorganized in some way. Assume that the train arriving from the direction A has N <= 1000 coaches numbered in increasing order 1, 2, ..., N. The chief for train reorganizations must know whether it is possible to marshal coaches continuing in the direction B so that their order will be a1, a2, ..., aN. Help him and write a program that decides whether it is possible to get the required order of coaches. You can assume that single coaches can be disconnected from the train before they enter the station and that they can move themselves until they are on the track in the direction B. You can also suppose that at any time there can be located as many coaches as necessary in the station. But once a coach has entered the station it cannot return to the track in the direction A and also once it has left the station in the direction B it cannot return back to the station.

**Input**

The input consists of blocks of lines. Each block except the last describes one train and possibly more requirements for its reorganization. In the first line of the block there is the integer N described above. In each of the next lines of the block there is a permutation of 1, 2, ..., N. The last line of the block contains just 0.   
  
The last block consists of just one line containing 0.

**Output**

The output contains the lines corresponding to the lines with permutations in the input. A line of the output contains Yes if it is possible to marshal the coaches in the order required on the corresponding line of the input. Otherwise it contains No. In addition, there is one empty line after the lines corresponding to one block of the input. There is no line in the output corresponding to the last ``null'' block of the input.

**Sample Input**

5

1 2 3 4 5

5 4 1 2 3

0

6

6 5 4 3 2 1

0

0

**Sample Output**

Yes

No

Yes