Indian Institute of technology, Guwahati

Department of Computer Science and Engineering

Data Structure Lab: (CS210)

Offline Assignment: 1

Date: 7th August, 2016. **Total Marks: 20**

Deadline: 10PM, 13th August, 2017. (Hard Deadline)

1. Write a function that search a number in a single linked list. Your list may contain **cycle**. Create a linked list with inputs given as below format.
   <**number1 number2 number3**>. You need to insert the number2 in between number1 and number3. number2 is always greater than 0. number1 is 0 indicates insert at start. number3 is 0 indicates insert at the end. -1 indices end of insertions. **[10]**
   1 0 5 1: insert 5 at start. Here 1 is redundant
   2 1 5 0: insert 5 at the end. Here 1 is redundant.
   3 1 2 3: insert 2 in between 1 and 3. 1 and 3 may not be adjacent nodes.
   4 -1: end of insertion.

   Input: sequence of <**number1 number2 number3**> tuple followed by a number to be searched.

   Output: FOUND/NOT FOUND

   Test1:
   0 1 1
   0 2 1
   0 3 1
   0 8 1
   2 7 1
   1 9 0
   -1
   1
   Output: FOUND

   Test2:
   0 4 1
   0 3 1
   0 2 1
   0 1 1
   1 5 0
   1 6 0
   6 7 2
   -1
   10
   Output: NOT FOUND

Test3:

```
0 4 1
0 3 1
0 2 1
0 1 1
1 5 0
1 6 0
6 7 1
-1
10
Output: NOT FOUND
```

2. We have two queues, the input queue Q1 and the output queue Q2, and one stack S. We are only allowed to dequeue from Q1 and we are only allowed to enqueue into the output queue Q2. We are allowed to both push into and pop from the stack.

   Consider that we have the numbers 1 2 3 4 5 6 enqueued in Q1. Suppose we perform the following sequence of operations.

   ```
   enqueue(Q2, dequeue(Q1))
   push(S, dequeue(Q1))
   enqueue(Q2, dequeue(Q1))
   push(S, dequeue(Q1))
   enqueue(Q2, pop(S))
   enqueue(Q2, pop(S))
   enqueue(Q2, dequeue(Q1))
   enqueue(Q2, dequeue(Q1))
   ```

   then the output queue contains 1 3 4 2 5 6

   This is an example of what we call a *stack permutation* i.e. A stack permutation is a ordering of numbers from 1 to n that can be obtained from the initial ordering 1, 2, ... n by a sequence of stack operations as described above.

   To clarify this, note that 1 5 3 4 2 6 is *not* a stack permutation. Intuitively this is because to enqueue 5 into Q2 after 1 we would have to push 2 3 4 into the stack which would then be output in the order 4 3 2, not in the order 3 4 2.

   In this assignment you will be given a permutation of n numbers and asked to check if it is a stack permutation or not. The TA will give you the number n as input and will give you a permutation. If it is a stack permutation your program will have to return the sequence of operations that formed that permutation. If it is not a permutation, your program will have to say it is not a permutation, and will have to give the reason why (as given above).

[10]