Data Structures Lab (CSC204) **Session: Monsoon 2023-24**
III BTech (CSE) **Date: 04.09.2023**

Assignment-5 (Stack-II and Queue)

---

1. Stack is a well-known linear data structure which follows Last-In-First-Out (LIFO) order for insertion and deletion of the elements. Create a stack ADT which supports $CreateS(), Push(), Pop(), Top()$, and $isEmpty()$ operations. Solve the following problems using the created stack ADT. **Do not use C++ STL.**

a) Suppose you have two arrays, $pushed$ and $popped$, where $pushed$ represents the elements to be pushed into a stack in a specific order, and $popped$ represents the desired order in which the elements should be popped out. Write a program to determine whether there exists any valid sequence of $push$ and pop operations that would generate the popped array from the pushed array. If valid sequence of operations exists then print that sequence.

Sample Input:
Enter the number of elements: 5
Enter the entries of array $pushed$: 1, 2, 3, 4, 5
Enter the entries of array $popped$: 4, 5, 3, 2, 1

Sample Output:
True
The valid sequence:
$push(1), push(2), push(3), push(4), pop(4), push(5), pop(5), pop(3), pop(2), pop(1)$

b) Write a program to populate a stack with N numbers, and then sort the elements of the stack in descending order. You are allowed to use another additional stack for this task, but you must not use any other data structure besides stacks.

Sample Input:
Enter the size of stack: 5
Enter the stack elements in order: 4 1 3 2 5

Sample Output:
Sorted stack is: 5 4 3 2 1

c) Write a function $\boldsymbol{ToPostfix}()$ to covert a given infix expression to its equivalent postfix expression.

Sample Input:
Enter the infix expression: $2 + 3 * (4\text{^}2 - 8)\text{^}(2 + 1 * 2) - 7$

Sample Output:
Postfix expression: $2342\text{^}8 - 212 * +\text{^} * +7 -$

2. A circular queue is a specific type of queue where the last element of the queue is connected with the first element of the queue forming a circle. Write a program to create an array-based ADT of circular queue of size N which will support $Enqueue$ and $Dequeue$ operations. Then solve the following problem using the queue ADT you have created.

Suppose you have a circular queue that initially contains integer numbers. Your task is to process these numbers one by one according to the below rules:

(i) At each step, you process the number at the front of the circular queue.

(ii) After processing, this number is re-enqueued into the circular queue, but its value is decreased by 1.

(iii) If a number's value becomes 0 after decrementing by 1, it is not re-enqueued and is removed from the circular queue.

(iv) The next number to be processed is the one that follows the previously processed number in the circular queue.

Write a program that simulates this process and prints the numbers in the order in which they are processed.

Sample Input:
Enter the number of elements: 4
Enter the numbers: 4 1 3 2

Sample Output: 4 1 3 2 3 2 1 2 1 1