Question - 1

Stack, Queue,Bag and List

SCORE: **5 points**

A data buffer is a region to temporarily store data while it is being moved from one place to another, Typically, the data is stored in a buffer as it is retrieved from an input device or just before it is sent to an output device. which of the following data structure is better in this situation?

- ( ) Stack
- (•) Queue
- ( ) Bag
- ( ) ArrayList

Question - 2

Stack and Queue

SCORE: **5 points**

Suppose that you have empty two data structures: a stack and a queue. Items can be pushed on to the stack and, at any time, the item on the top of the stack may be popped and enqueued into the queue.

The following items: *a, b, c, d, e,* and *f* are pushed onto the stack in the order given. Which of the following sequences can *never* be the state of the queue after the stack is empty? Note that the left-most item shown is the oldest element of the queue.

- ( ) f e d c b a
- ( ) b c a f e d
- ( ) d c e f b a
- (•) c a b d e f

Question - 3

Validate Palindrome

SCORE: **20 points**

Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring case. A palindrome is a string which reads the same forwards as backwards.
Note: For the purpose of this problem, we define an empty string as a valid palindrome. All inputs are lower cases.

Question - 4

SCORE: **20 points**

Implementation

There is a deck of N cards on the desk and you are going to play a game: take out the card on the top and record it, then put the next card to the bottom of the deck. repeat this process until the deck is empty. The result is the record data.
Please implement a queue to simulate this process. The input provided will contain:
(a) the capacity of the queue, followed by a comma (",") and
(b) the order of the deck, comma-separated and enclosed in "{}"
The size of the deck is no larger than the capacity of the queue. You should return an array which contains the record of this game.
Sample:
input: 10, {1,2,3,4,5}
Process: {1,2,3,4,5} -> {3,4,5,2} -> {5,2,4} -> {4,2} -> {2} Queue
{1} -> {1,3} -> {1,3,5} -> {1,3,5,4} -> {1,3,5,4,2} ResultArray
output: {1,3,5,4,2}