

- 1) Implement LinkedList data structure and its operations in java —Possible points 12

Operations to implement:

- a) size
- b) isEmpty
- c) addNodeAtHead
- d) addNodeAtTail
- e) addNodeAt(Nth)Position
- f) reverseALinkList
- g) getHeadNode
- h) getTailNode
- g) deleteNodeAt(nth) position

For each operation write test coverage to validate your implementation. Also state the time complexity and space complexity for each operation in your report.

- 2) Implement Stack data structure and its operations in java using LinkedList data structure from question 1. —Possible points 12

Operations to implement:

- a) push
- b) pop
- c) peek
- d) size
- e) isEmpty

For each operation write test coverage to validate your implementation. Also state the time complexity and space complexity for each operation in your report. report.

- 3) Implement Queue data structure and its operations in java using LinkedList data structure from question 1. —Possible points 12

Operations to implement:

- a) enqueue
- b) dequeue
- c) peek
- d) isEmpty
- e) size

For each operation write test coverage to validate your implementation. Also state the time complexity and space complexity for each operation in your report.

- 4) Implement Double-LinkedList data structure and its operations in java —Possible points 12

Operations to implement:

- a) size
- b) isEmpty
- c) addNodeAtHead
- d) addNodeAtTail
- e) addNodeAt(Nth)Position
- f) reverseALinkList
- g) getHeadNode
- h) getTailNode
- i) deleteNodeAt(nth) position

For each operation write test coverage to validate your implementation. Also state the time complexity and space complexity for each operation in your report.

5) Can we implement a Stack data structure using arrays? Give reasons(pro/cons) if yes/no? —

**Possible points 2**

6) Implement a Stack Data structure using Queue data structure from question 3? (Java code)

**—Possible points 10**

7) Implement a Queue Data Structure using Stack Data Structure from question 2 ? (Java Code) **—Possible points 10**

a) How many stacks do you need to implement?

b) Implement multiple approaches if it's possible?

8) Implement a java code to check whether a given linked list is circular ? Use LinkedList Data Structure from question 1? **—Possible points 10**

9) Implement a java code to check whether an ASCII string has unique characters? Your code should preferably run in  $O(n)$  time and use only  $O(1)$  memory? **—Possible points 10**

10) Given sequence consisting of parentheses, determine whether the expression is balanced or not. A sequence of parentheses is balanced if every open parentheses can be paired uniquely with a closed parentheses that occurs after the former. Also, the interval between them must be balanced. You will be given three types of parentheses: (, {, and [.

**—Possible points 10**

{[()] } - This is a balanced parentheses.

{[( )]} - This is not a balanced parentheses.

Constraints :

$1 \leq \text{lens} \leq 1000$ , where lens is the length of the sequence.

Each character of the sequence will be one of {, }, (, ), [, ].

Which is the ideal data structure can be used? Implement a java code and test cases for it?