University of Kelaniya – Sri Lanka
Centre for Distance & Continuing Education
Bachelor of Science (General) External
Second year First semester examination - 2024
Faculty of Science
COSC 26563 -Data Structures and Algorithms

No. of Questions: **Four (04)** No. of Pages: **Three (03)** Time: **2 Hours and 30 Minutes**

Answer <u>ALL</u> questions.

---

1. a) Explain the difference between *Linear and Non-Linear* data Structures. Give two (02) examples for each. **(12 marks)**

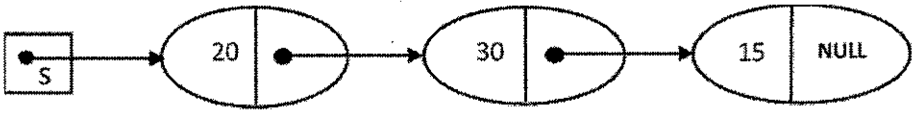
b) Using a real-world application, briefly explain the concept of the *stack* data structure and how it operates based on the LIFO (Last-In-First-Out) principle. **(12 marks)**

c) Defining necessary datatypes and constructors, give a suitable structure in Java language to represent a *stack* using the contiguous implementation. **(12 marks)**

d) Using the datatypes and constructors defined in part 1 c), write functions in Java to accomplish the following. **(28 marks)**

(i) To check whether the given stack is empty or not.
(ii) To check whether the given stack is full or not.
(iii) To pop an element from the stack.
(iv) To push an element onto the stack.

e) Consider the following node and arc diagram of a stack of integers 'S'.



(i) Modify the above node and arc diagram to show what happens during a pop operation. **(12 marks)**

(ii) Modify the node and arc diagram you got for part 1 e) (i) above to show what happens when an element with value 32 is pushed into the stack. **(12 marks)**

(iii) Show the steps associated with destroying the stack using node and arc diagrams. **(12 marks)**

2. a) Using an example, briefly describe the concept of the *Queue* data structure. **(15 marks)**

b) Discuss the similarities and the differences between *Stack* and *Queue* data structures. **(10 marks)**

c) Defining necessary datatypes and constructors, give a suitable class structure in Java language to represent a *queue* using the linked implementation. **(10 marks)**

d) Using the datatypes and constructors defined in part 2 c), write functions in Java to accomplish the following:

(i) To check whether the given *Queue* is empty or not. **(10 marks)**

(ii) To insert an element to the *Queue*. **(10 marks)**

(iii) To remove an element from the *Queue*. **(10 marks)**

(iv) To get the size of a given *Queue*. **(10 marks)**

e) Students in a class have marks for three subjects: Mathematics, Chemistry, and Physics. Using a Queue, write a Java program to read the marks and print only the average marks of each student in the order in which the data was inserted. **(25 marks)**

3. a) Briefly describe the properties of a List data structure. **(15 marks)**

b) Defining necessary datatypes and constructors, give a suitable class structure in Java language to represent a list data structure using the contiguous implementation. **(15 marks)**

c) Using the classes and datatypes defined in part 3 b), Write functions in Java language to accomplish the following:

(i) To check whether the given list is empty or not. **(10 marks)**

(ii) To check whether the given list is full or not. **(10 marks)**

(iii) To insert an element to the last position in the list. **(10 marks)**

d) With the aid of node and arc diagrams, briefly explain the steps to be followed when inserting a new element into a list, if the position to be inserted is given. **(20 marks)**

e) Write a function using the Java language to implement the iterative version of the binary searching algorithm. **(20 marks)**

4. a) (i) Define *Binary tree* used in data structures. **(05 marks)**

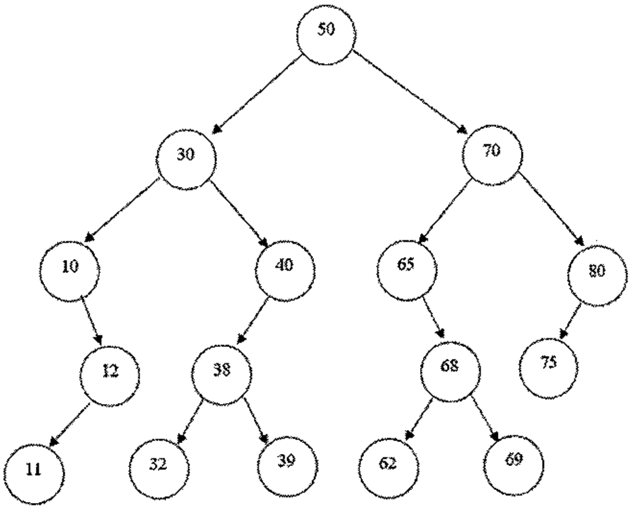
b) (i) Briefly explain the three traversal orders of *binary trees*. **(15 marks)**

(ii) Consider the following *binary tree* and write the sequence of the integers as an outcome, with respect to the following traversal order. **(30 marks)**

I. Preorder

II. Inorder

III. Postorder



c) Draw the expression tree that represents the following arithmetic expression. **(10 marks)**
((x + y) * (a - b)) / (m + (n * p))

d) (i) Define a binary search tree. **(10 marks)**

(ii) By using a diagram, show the main steps to insert the following keys into an initially empty binary search tree. **(10 marks)**

L, T, G, R, V, W, K, A, H

e) Consider the following definition for the linked implementation of a *binary tree.*

```
public class Node {
    Node left, right;
    int data;
    public Node(int n)
    {
        left = null;
        right = null;
        data = n;
    }
}
```

Using the above definition, write functions in Java language to accomplish the following tasks.

(i) To initialize a tree **(05 marks)**

(ii) To implement the preorder traversal **(05 marks)**

(iii) To implement the inorder traversal **(05 marks)**

(iv) To implement the postorder traversal **(05 marks)**