PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

END SEMESTER ASSESSMENT (ESA) - DEC 2023

UE22CS252A. - Data Structures and its Applications

Total Marks : 100.0

1.a. Convert the given infix expression **((A+B)-C*(D/E))+F** to postfix and prefix expressions, and then evaluate the expressions using the provided values:
Given values: A=8, B=3, C=4, D=18, E=6, F=7 (4.0 Marks)

1.b. Design a singly linked list that can store elements in either ascending or descending order based on user preference. Implement a function, **void ordered_ins(int x) { }**, which inserts a given element into the linked list while maintaining the specified order. (6.0 Marks)

1.c. What does the following code snippet do? Explain the functionality implemented in the given code.
Consider the following Node structure for doubly linked list

```
struct Node {
   char data;
   struct Node* next;
   struct Node* prev;
};

int isWHATTT(struct Node* head)
{
   struct Node *start = head, *end = head;
      while (end->next != NULL)
    {        end = end->next;     }

   while (start != end && start->prev != end)
    {
      if (start->data != end->data) {
         return 0;
      }
      start = start->next;
      end = end->prev;
   }
   return 1;
}
```

(4.0 Marks)

1.d. Complete the following code to PUSH the elements onto the STACK and POP the elementsfrom the STACK. Check the boundary conditions as well.

```
void push(int s[],int *t, int x)
{
//TO DO
}

int pop(int s[],int *t)
{
//TO DO
}
```

(6.0 Marks)

1.e. Explain the following each with an example.
1. Dynamic Memory Allocation
2. Applications of STACKS.

(5.0 Marks)

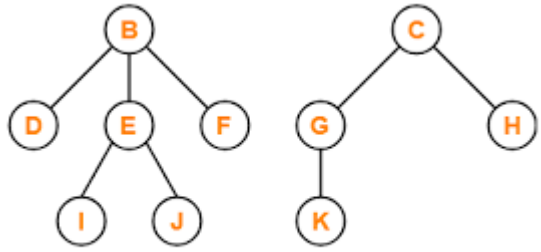
2.a. Explain the concept of a circular queue. What problem does it solve, and how does it overcome the limitations of a regular linear queue? Implement a circular queue and perform the following operations efficiently:

- Enqueue multiple elements in a loop.
- Dequeue elements until the queue is empty.
- Display the elements of the circular queue at any point during the operations. (9.0 Marks)

2.b. Consider the following sequence of elements.
Given sequence: [8, 3, 10, 1, 6, 14, 4, 7, 13]
Construct a binary search tree and traverse the same in PREORDER and POSTORDER techniques. (6.0 Marks)

2.c. Convert the following Forest to a Binary Tree.

(4.0 Marks)



Forest

2.d. Write recursive functions to perform the following operations on a binary search tree.

i. Find the number of nodes in a BST.

ii. Search for an element in a BST.

(Note: write only the functions).

```
void no_of_nodes(TREE *root)
{
//TO DO
}
void bstsearch(TREE *root,int key)
{
  //TO DO
}
```

(6.0 Marks)

3.a. You are tasked with implementing a function to create an expression tree from a given postfix expression. The expression may consist of variables (single lowercase letters) and binary operators ('+', '-', '*', '/'). The goal is to construct a binary tree that represents the hierarchical structure of the expression.
Write a C function named create_exptree that takes a postfix expression as input and returns the root of the expression tree. The function should adhere to the following specifications:
struct Node* createExpressionTree(char exp[]);
postfixExpression: A null-terminated string representing a postfix expression. The expression contains single lowercase letters representing variables and binary operators
('+', '-', '*', '/').

```
TREE* create_exptree(TREE *root, char exp[])
{ TREE *temp,*new;
  char ch;
  int i=0;
    while(exp[i]!='\0')
    {
    ch=exp[i];
    new=(TREE *)(malloc(sizeof(TREE)));
    new->info=ch;
    new->left=NULL;
    new->right=NULL;

       // TODO
    }
    return new;
}
```

(5.0 Marks)

3.b. Explain the following with an example:
1. Representation of Graphs
2. Priority Queues using Heaps.

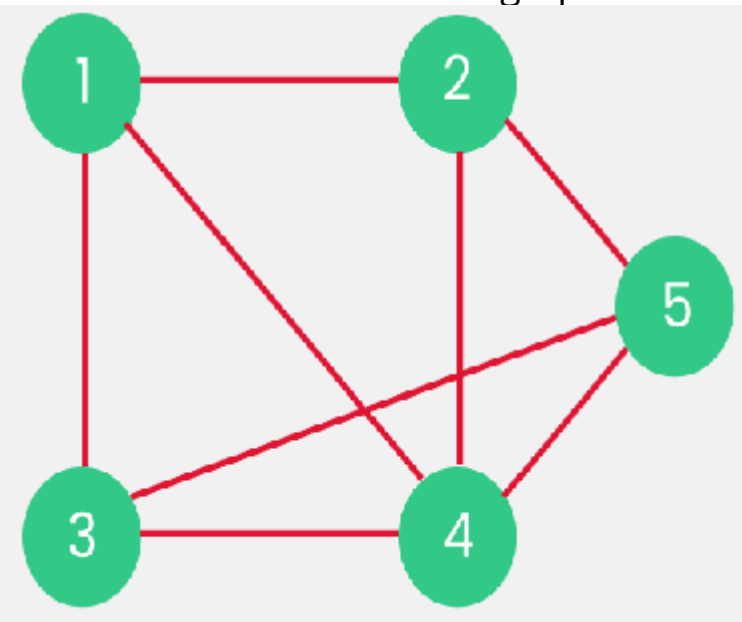
(8.0 Marks)

3.c. Create an AVL (Adelson-Velsky and Landis) Tree from a given sequence of elements.
[10, 5, 15, 2, 7, 12, 20].
Perform the following operations on the tree step by step an balance the tree to create an AVL tree.

1. Insert 9, 30 and 45.
2. Now insert 18 and 16 respectively.

Write the final balanced AVL tree. (6.0 Marks)

3.d. Consider an undirected graph as shown below.



Traverse the graph using Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms. Let the start vertex be 4.
What data structure is used in either of the algorithms? (6.0 Marks)

4.a. Assume initially empty B-tree.
Perform the following operations in the given order.
i. Construct a B-Tree of order 3 for the following elements entered in the order as given below : 78, 52, 81, 40, 33, 90, 85, 20 and 38.
ii. Insert 100 , 120 and 140 into the tree.
iii. Delete 90 from the tree.
Write the final B tree of order 3. (6.0 Marks)

4.b. Build a max heap for the following list of elements:
12, 35, 87, 26, 9, 28, 7.
Perform the following operations:
Delete the maximum element one after the other and now show the maxheap . (5.0 Marks)

4.c. Construct a trie tree for the following strings:
That, There, This, Does, Did. (4.0 Marks)

4.d. Draw the 11-entry hash that results from using the hash function h(i) = 2i mod 11 to hash keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5. Show the hash table when collisions are handled by using
(i) Separate Chaining
(ii) Linear Probing.
Where' i' is the key element used in the hash function. (10.0 Marks)