CONFIDENTIAL

UNIVERSITY COLLEGE TATI (UCTATI)

FINAL EXAMINATION QUESTION BOOKLET

COURSE CODE	: DCT2023
COURSE	: DATA STRUCTURE AND FILE PROCESSING
SEMESTER/SESSION	: 1/2024-2025
DURATION	: 3 HOURS

Instructions:

1. This booklet contains 5 questions. Answer ALL questions.
2. All answers should be written in answer booklet.
3. Write legibly and draw sketches wherever required.
4. If in doubt, raise your hands and ask the invigilator.

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO

THIS BOOKLET CONTAINS 6 PRINTED PAGES INCLUDING COVER PAGE

QUESTION 1

a) Define the term **Queue** in data structure. (2 marks)

b) List **THREE (3)** applications of a **Queue**. (3 marks)

c) Explain **TWO (2)** differences between a **Queue** and a **Stack** in data structures. (6 marks)

QUESTION 2

The C++ program in Figure 1 demonstrates basic **Queue** operations such as adding, deleting, and retrieving the front and back values of a **Queue**. Questions a) to e) are based on this program.

```
#include <cstdlib>
#include <iostream>
using namespace std;
#define MAX 5
int queue_array[MAX];
int front;
int back;

void initializeQueue(){
      front = - 1;
      back = - 1;
}

void addQueue(int add_item){
    if (front == - 1)
      front = 0;

    back = back + 1;
    queue_array[back] = add_item;
}


void deleteQueue(){
    front = front + 1;
}

int getFront(){
      return  queue_array[front];

}

int getBack(){
      return  queue_array[back];
}

int main(int argc, char *argv[])
{
    initializeQueue(); //statement i
      addQueue(10); //statement ii
      addQueue(30); //statement iii
      addQueue(70);
      deleteQueue();
      cout<<getFront()<<endl;
      cout<<getBack()<<endl;
    return 0;

}
```

Figure 1

a) Identify the size of the **Queue** defined in the program. (2 marks)

b) Explain how the **addQueue** function in the program perform the add operation in the `queue_array`. (4 marks)

c) Explain how the **deleteQueue** function in the program perform the delete operation in the `queue_array`. (4 marks)

d) Describe the purpose of **getFront** function in the program. (2 marks)

e) Explain how **initializeQueue** function in the program perform the initialization process in the **Queue**. (4 marks)

f) Write the data stored in the variables **front, back** and **queue_array** when each of the statement from i. to iii. is executed. Complete the following Figure 2. (6 marks)

front	back

queue _array[4]	
queue _array[3]	
queue _array[2]	
queue _array[1]	
queue _array[0]	

`initializeQueue(); //statement i`

front	back

queue _array[4]	
queue _array[3]	
queue _array[2]	
queue _array[1]	
queue _array[0]	

`addQueue(10); //statement ii`

front	back

queue _array[4]	
queue _array[3]	
queue _array[2]	
queue _array[1]	
queue _array[0]	

`addQueue(30); //statement iii`

Figure 2

QUESTION 3

a) Define the term **Sorting**? (2 marks)

b) Give the advantage and disadvantage of **Selection Sort.** (4 marks)

c) List **THREE (3)** sorting techniques other than **Selection Sort.** (3 marks)

d) Identify the name of the following sorting algorithm. (2 marks)

1. *Compare the first element in the input array with the next element.*
2. *If the first element is greater, swap the two elements; otherwise, move to the next pair of elements in the array.*
3. *Repeat Step 2 until you reach the end of the array.*
4. *Check if the array is sorted; if not, repeat the process (Steps 1 to 3) from the last element of the array back to the first.*

QUESTION 4

a) Sort the following list using **Selection Sort**. Show the list after each iteration of the outer for loop. (7 marks)

26, 45, 17, 65, 33, 55, 12, 18

b) Given an **array A** of 20 non-negative integers, sort the array in ascending order using **Bubble Sort** and print the sorted array. Complete the following C++ code. (15 marks)

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
const int LENGTH= 20;
int
A[LENGTH]={6,1,20,2,11,3,4,5,13,7,16,10,8,9,12,15,19,17,1
8,14,};

//write your code here


 return 0;
}
```

QUESTION 5

a) Define a **Binary Tree**? (3 marks)

b) List and briefly explain **THREE (3)** applications that use binary trees. (9 marks)

c) Draw **THREE** (3) different binary trees, each with three nodes. (6 marks)

d) Consider the binary tree in Figure 3 and answer the following questions.

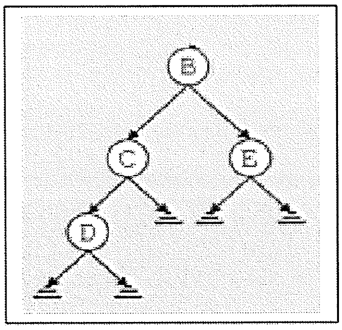

Figure 3

i) Find **LB**, the left subtree of B, and **RB**, the right subtree of B. (4 marks)

ii) List the nodes of this binary tree in **inorder**, **preorder**, and **postorder** sequences. (6 marks)

e) A function to print a binary tree in **inorder** sequence is given below, where '***p***' is a pointer to a **binaryTreeNode**. Modify the function to print the binary tree in **preorder** sequence. (6 marks)

```
void inorder(binaryTreeNode *p)
{
if (p != NULL)
    {
        inorder(p->llink);
        cout << p->info << " ";
        inorder(p->rlink);
    }
}
```

--End of question---------------------------------------