National University of Computer and Emerging Sciences, Lahore Campus

Course:	DS Lab	Course Code:	CL 2001
Program:	BS (Computer Science)	Semester:	Spring 2022
Duration:	2 Hours	Total Marks:	50
Paper Date:	2nd Jan, 2023	Weight:	45%
Section:	BCS-3A To BCS-3G	Page(s):	3
Exam:	Lab Final term	Reg. No.	

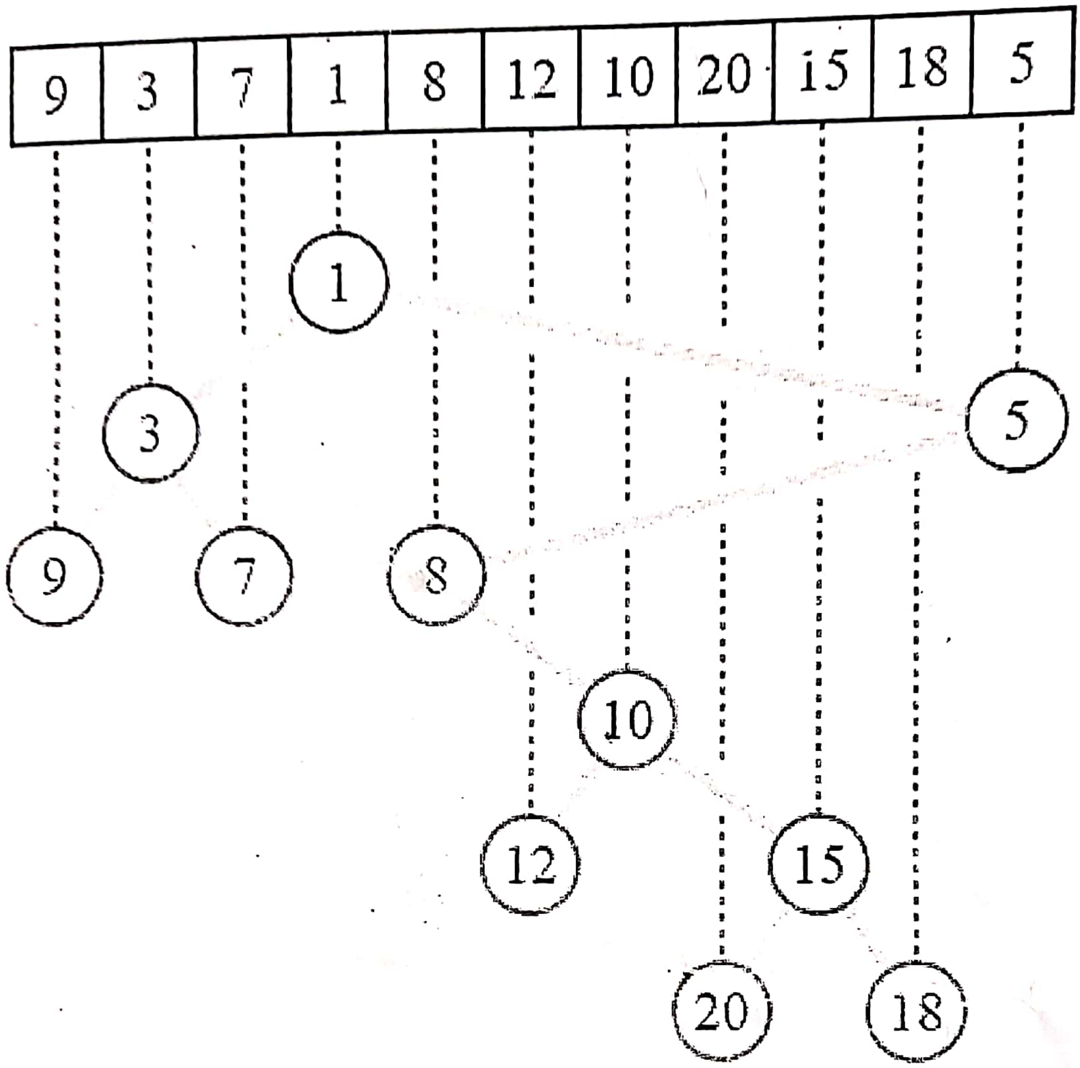
Read the below Instructions Carefully:

- Understanding the question statement is also part of the exam, so do not ask for any clarification. In case of any ambiguity, make suitable assumptions.
- You have to complete the exam in 2 hrs. No extra time will be given for submission.
- Submit a single **.cpp file** for each question named as **21L-1122 (Q#)**
- Place all .cpp files into the **folder** named as **21L-1122**
- Submit folder on **cactus** by following path: \\cactus1\ Xeon\ Fall 2022\ Mamoona Akbar\ DS\ Final\sec
- Your code should be **intended** and **commented** properly. Use **meaningful variable names.**
- It is your responsibility to save your code from being copied. All matching codes will be considered cheating cases. **PLAGIARISM** will result in forwarding of **case** to **Disciplinary Committee** and **negative marks** in Midterm.

Question 1: Marks 25

Write an efficient algorithm to construct a **Cartesian tree** from an array using **inorder traversal.** A Cartesian tree is a **binary tree** with the **heap property**: the parent of any node has a smaller value than the node itself.

For example, the following figure shows an example of a Cartesian tree derived from the sequence of numbers in inorder:



9 7 3

18 15

8

1 3 5 .

1 3 5 7 8 9

min 1.

3

arr[j]

3 $\boxed{9\ 7}$

while size 6=0

startIndex ++

Question 2: Marks 25

Given an M × N matrix of characters, find all occurrences of a given string in the matrix. We are allowed to search the string in all eight possible directions, i.e., North, West, South, East, North-East, North-West, South-East, South-West. Note that there should not be any cycles in the output path.

For example, consider the following matrix of characters,

$$\begin{array}{cc} 0 & \\ 1 & \\ 2 & \\ & \\ & \end{array}\begin{bmatrix} 'D' & 'E' & 'M' & 'X' & 'B' \\ 'A' & 'O' & 'E' & 'P' & 'E' \\ 'D' & 'D' & 'C' & 'O' & 'D' \\ 'E' & 'B' & 'E' & 'D' & 'S' \\ 'C' & 'P' & 'Y' & 'E' & 'N' \end{bmatrix}$$

0 1 2

0
1
2
3
4

0

If the given string is " **CODE**", following are all its occurrences in the matrix:

```
C(2, 2)  O(1, 1)  D(0, 0)  E(0, 1)
C(2, 2)  O(1, 1)  D(2, 0)  E(3, 0)
C(2, 2)  O(1, 1)  D(2, 1)  E(1, 2)
C(2, 2)  O(1, 1)  D(2, 1)  E(3, 0)
C(2, 2)  O(1, 1)  D(2, 1)  E(3, 2)
C(2, 2)  O(2, 3)  D(2, 4)  E(1, 4)
C(2, 2)  O(2, 3)  D(3, 3)  E(3, 2)
C(2, 2)  O(2, 3)  D(3, 3)  E(4, 3)
```

C

Note: used **DFS** to solve this problem.