**Data Structures and Algorithms, Fall 2020**

**Deadline: January 24, 2020, 11:59pm (sharp)**

**Group Policy: No groups allowed – to be submitted individually**

**PROBLEM STATEMENET:**

**Implement the following tree:**

* Red-Black Tree – cannot have duplicate values (you don’t even need to check it; I will not enter duplicate values while checking)
* Codes can be recursive OR iterative, your choice

**Structure of Node (you cannot change it):**

struct Node

{

int data;

Node\*leftChild;

Node\*rightChild;

char colour;

};

**Your project should have the following main menu:**

Press 1 to insert values one by one in the tree

Press 2 to delete a value from the tree **(bonus/optional)**

Press 3 for searching a value from the tree

Press 4 for pre-order traversal NLR

Press 5 for in-order traversal LNR

Press 6 for post-order traversal LRN

Press 7 for pre-order traversal 2 NRL

Press 8 for in-order traversal 2 RNL

Press 9 for post-order traversal 2 RLN

Press 10 to destroy the tree (all nodes must be deleted)

Press 11 to delete all values in the tree greater than X

**(bonus/optional)**

Press 12 for displaying parent of a node present in Tree

Press 13 to read integer values from the file “input.txt”

to create a red-black tree

Press 14 to EXIT

**The program should exit when option 14 from the main menu is selected.**

**Please note that in case of red-black tree, the colour of a particular node should also be displayed along with its value for options 3, 4, 5, 6, 7, 8, 9, 12**

**The non-empty “input.txt” will have the data in such a way that a new value will be placed on every new line. For example, the following file (containing 7 values) is valid for creating the red-black tree (there may be less or more than 7 values):**

10

16

2

-5

0

22

1024