BACHELOR OF COMPUTER SCIENCE
FACULTY/SCHOOL OF COMPUTER SCIENCE
BINA NUSANTARA UNIVERSITY
JAKARTA

ASSESSMENT FORM

Course: COMP6048001 - Data Structure

Method of Assessment: Case Study

Semester/Academic Year: 2/2022-2023

Name of Lecturer : Muhammad Fikri Hasani, S.Kom., M.T.

Date : 5 Juni 2023

Class : LC01

Topic : Review II

Group Members :	1 Jonathan Alvindo Fernandi 2_____ 3_____ 4_____ 5_____ 6_____ 7_____ 8_____

Student Outcomes:
SO 2 - Mampu merancang, mengimplementasikan, dan mengevaluasi solusi berbasis komputasi untuk memenuhi serangkaian persyaratan komputasi dalam konteks ilmu komputer
Able to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of computer science

Learning Objectives:
LObj 2.2 - Mampu mengimplementasikan solusi berbasis komputasi untuk memenuhi serangkaian persyaratan komputasi tertentu dalam konteks ilmu komputer
Able to implement a computing-based solution to meet a given set of computing requirements in the context of computer science

Learning Outcomes:
LO 3 - Apply data structures using C

No	Related LO-LObj-SO	Assessment Criteria	Weight	Excellent (85 - 100)	Good (75 - 84)	Average (65 - 74)	Poor (0 - 64)	Score	(Score x Weight)
1	LO3-LObj 2.2 -SO 2	Ability to identify the problems to find the solution	**10%**	Able to identify both the input and output in full detail for the problem	Able to identify both the input and output but in less detail for the problem	Able to identify both the input and output but not in clear detail for the problem	Able to identify only the input or output of the problem		
2	LO3-LObj 2.2 -SO 2	Ability to design an algorithm for the problem	**40%**	Able to design an algorithm for the problem in full detail	Able to design an algorithm but not have full detail	Able to design an algorithm for the problem but not have a clear process	Able to design an algorithm that cannot be implemented in the problem		
3	LO3-LObj 2.2 -SO 2	Ability to solve the problem	**50%**	Able to solve 76% - 100% of the problem with a fully functional feature	Able to solve 51% - 75% of the problem and lack some features	Able to solve 26% - 50% of the problem and lack some features	Able to solve less than 25% of the problem		
		Total Score: ∑(Score x Weight)							

Remarks:

_____

_____

_____

_____

ASSESSMENT METHOD

Instructions

- This case study is individual with 1 week of processing.
- Design an algorithm in pseudocode/code to solve the problem and write down the algorithm.
- If the case study does not have specific instructions about the algorithm that must be used, it means that students can determine the best algorithm to solve the given problem.
- The report must be submitted in .pdf format to BINUSMAYA.

Note for Lecturers:

- Lecturers notify this case study to the student from Week 1.
- The deadline for the case study is one week after lecturers post it on BINUSMAYA.
- Students should submit the report to BINUSMAYA no later than the deadline.
- If the students do plagiarism, their score for the case study will be zero.

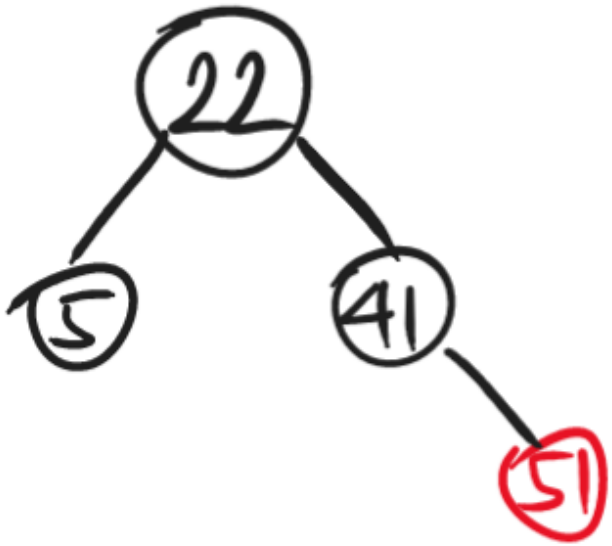
1. **RED BLACK TREE**
    a. (Bobot 10%, SO 2, LObj 2.2, LO 3) Create a Red Black Tree using the following sequence: 41, 22, 5, 51, 48, 29, 18, 21, 45, 3!
    Red Black Tree:

Insert(41)

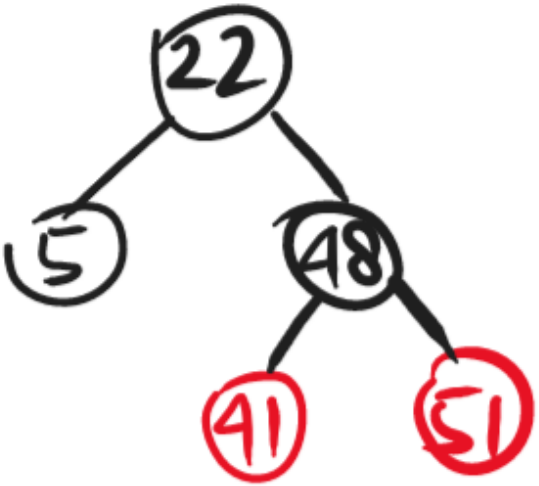
Insert(22)

Insert(5)

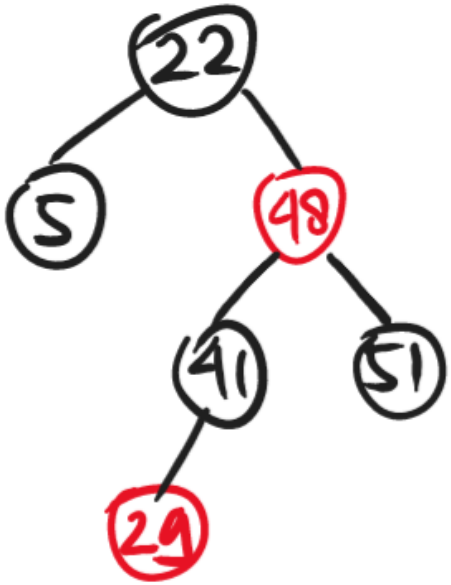
Insert(51)



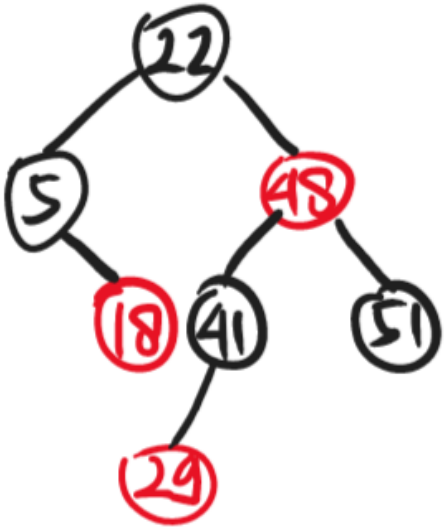
Insert(48)



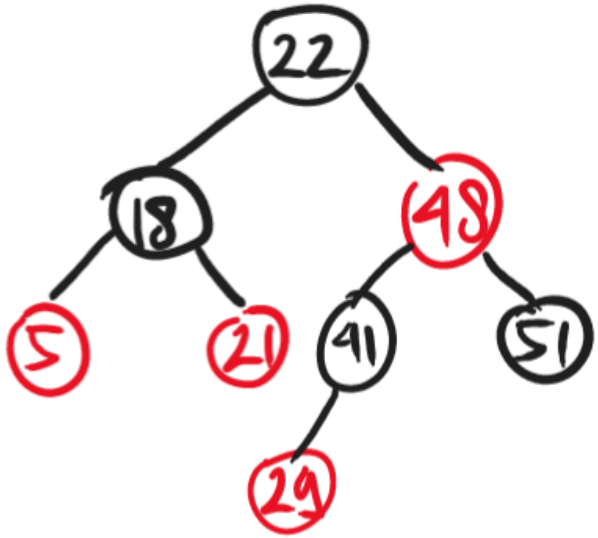
Insert(29)



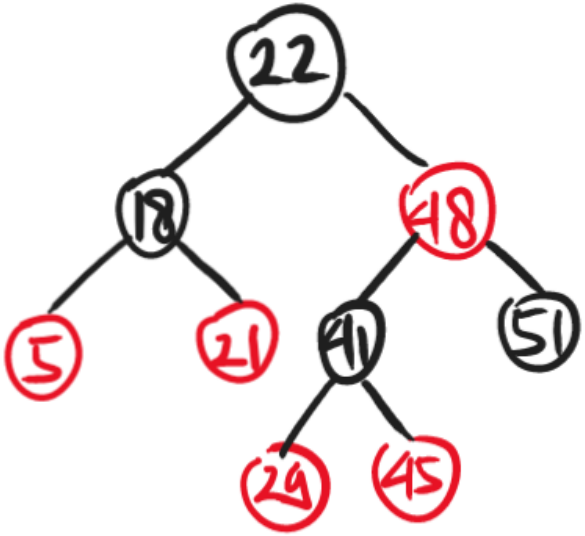
Insert(18)



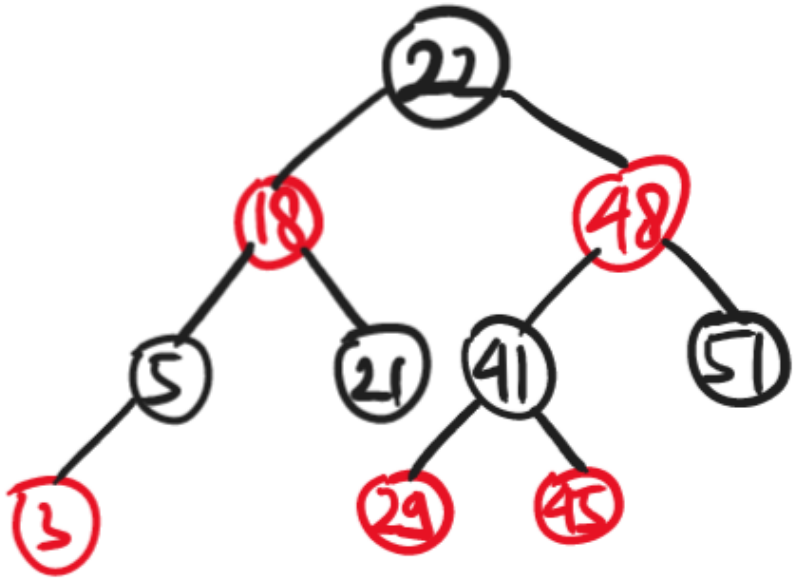
Insert(21)



Insert(45)



Insert (3)



b. (Bobot 20%, SO 2, LObj 2.2, LO 3) Try implementing Red Black Tree (a) insertion in C Program and print those data using Inorder Traversal!

```
Inoder Traversal of Created Tree
3 5 18 21 22 29 41 45 48 51
--------------------------------
Process exited after 0.1592 seconds with return value 0
Press any key to continue . . .
```

2. AVL TREE

a. Into an empty AVL Tree:

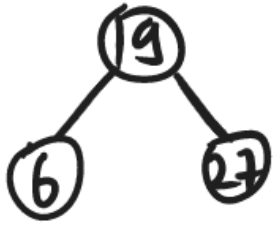
i. (Bobot 10%, SO 2, LObj 2.2, LO 3) Insert the following values: 6, 27, 19, 11, 36, 14, 81, 63, 75!

AVL Tree:

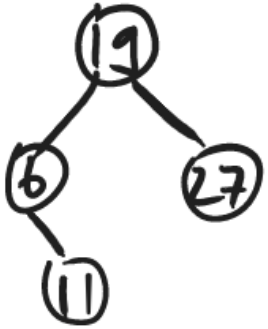
Insert (6)

Insert (27)

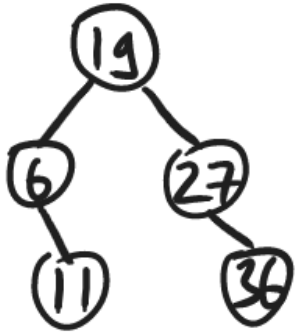
Insert (19)



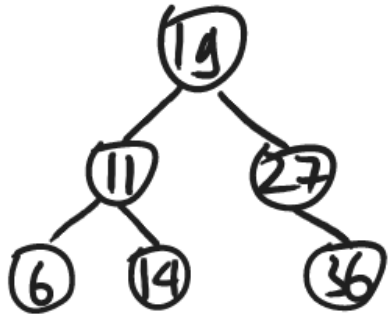
Insert (11)



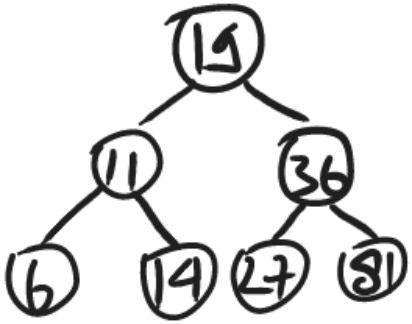
Insert(36)



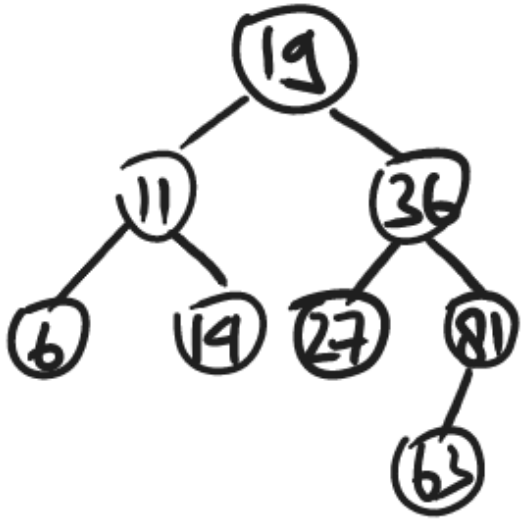
Insert(14)



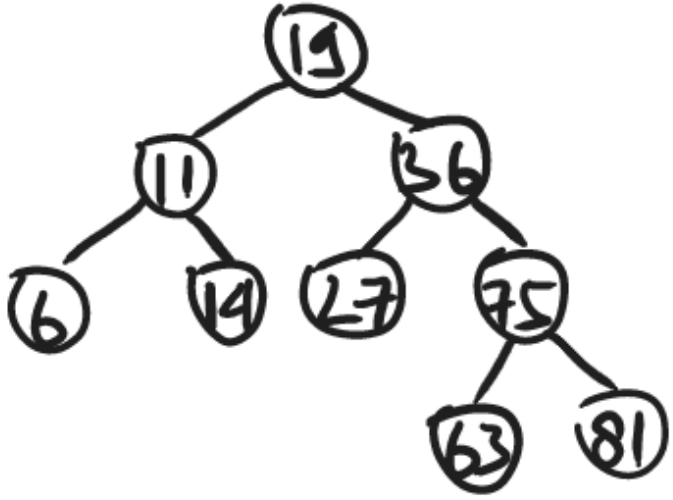
Insert(81)



Insert(63)

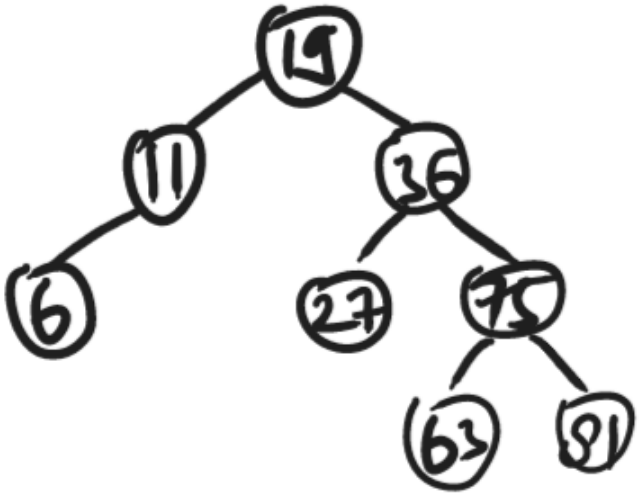


Insert(75)

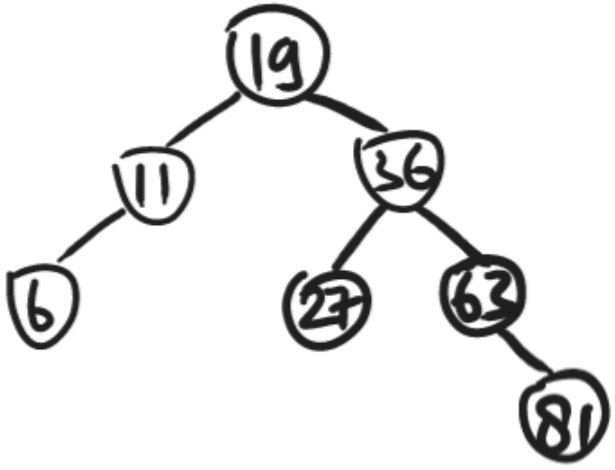


ii. (Bobot 10%, SO 2, LObj 2.2, LO 3) Delete the following values: 14, 75, 36, 19, 11!
AVL Tree:

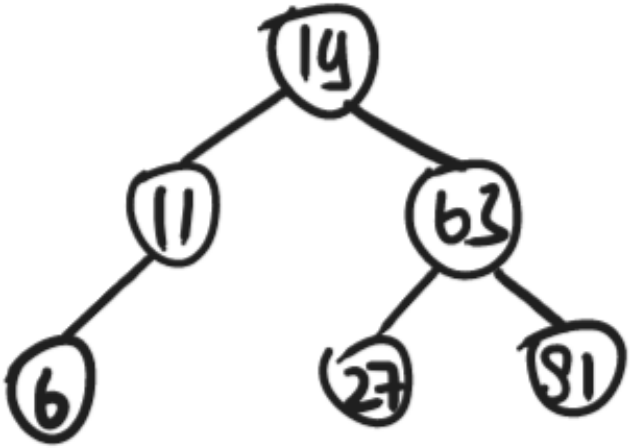
Delete(19)



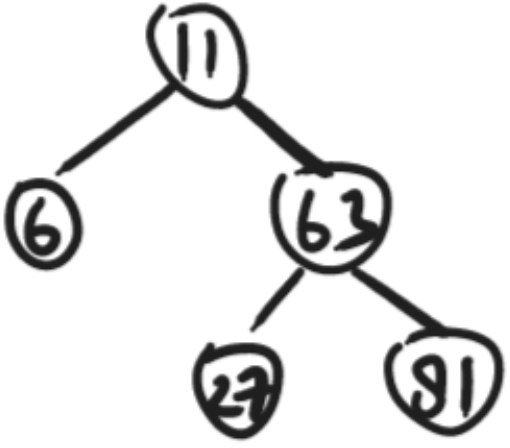
Delete(75)



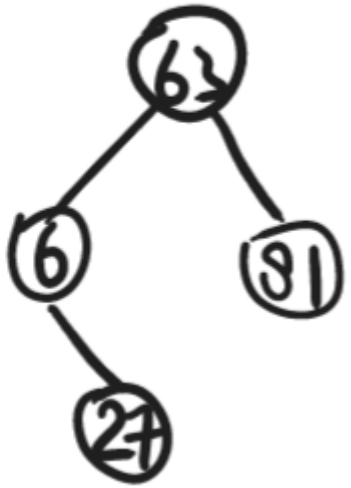
Delete(36)



Delete(14)



Delete(11)



b. (Bobot 50%, SO 2, LObj 2.2, LO 3) Write a program to insert, delete and print data from AVL Tree insertion in C Program!

```
1. Insertion
2. Deletion
3. Traversal
4. Exit
Choose:
```

i. Insertion

In this menu, the program will ask for the value that the user wants to insert into the AVL tree. Do the insertion in (a).

```
1. Insertion
2. Deletion
3. Traversal
4. Exit
Choose: 1
Insert: 6
```

ii. Deletion

In this menu, the program will ask for the value that the user wants to delete from the AVL tree. If the value is found in the tree, then it will be deleted, otherwise, the program will display the message ‘Data not found’. Do the deletion in (a).

```
1. Insertion
2. Deletion
3. Traversal
4. Exit
Choose: 2
Delete: 14
Data Found
Value 14 was deleted
```

```
1. Insertion
2. Deletion
3. Traversal
4. Exit
Choose: 2
Delete: 7
Data not found
```

iii. Traversal

In this menu, the program will print all data from the AVL tree in preorder, inorder and postorder.

```
1. Insertion
2. Deletion
3. Traversal
4. Exit
Choose: 3
Preorder: 19 11 6 14 36 27 75 63 81
Inorder: 11 6 14 19 36 27 75 63 81
Postorder: 11 6 14 36 27 75 63 81 19
```

iv. Exit

```
1. Insertion
2. Deletion
3. Traversal
4. Exit
Choose: 4
Thank you
```