**BACHELOR OF COMPUTER SCIENCE**

**FACULTY/SCHOOL OF 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 : ………………………**

**Date : ………………………**

**Class : ………………………**

**Topic : Review II**

|  |  |
| --- | --- |
| **Group Members :** | 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  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 computer**

***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 computer**

***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 fully 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 for 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 clear process | Able to design an algorithm but cannot be implemented in the problem |  |  |
| 3 | LO3-LObj 2.2 -SO 2 | Ability to solve the problem | **50 %** | Able to solve 76- 100% the problem with fully functional feature | Able to solve 51-75% 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 processing
* Design an algorithm in pseudocode/code to solve the problem and write down the algorithm
* If in the case study does not have specific instruction about the algorithm that must be used, it means that students can determine the best algorithm to solve the given problem
* Report will be submitted in pdf format to binusmaya

**Note for Lecturers**:

* The lecture notifies this case study to the student from Week 1
* Deadline for the case study is one week after the lecture post it on binusmaya
* The student should submit the report to binusmaya no later than deadline
* If the student do plagiarism, their score for this case study will be zero

1. **RED BLACK TREE**
   1. (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
   2. (Bobot 20%, SO 2, LObj 2.2, LO 3) Try implementing (a) Red Black Tree insertion in C Program and print those datas with InOrder Traversal

Graphical user interface, text, application

Description automatically generated

1. **AVL TREE**
   1. Into empty AVL Tree:
      1. (Bobot 10%, SO 2, LObj 2.2, LO 3) Insert the following values: 6, 27, 19, 11, 36, 14, 81, 63, 75
      2. (Bobot 10%, SO 2, LObj 2.2, LO 3) Delete the following values: 14, 75, 36, 19, 11
   2. (Bobot 50%, SO 2, LObj 2.2, LO 3) Write a program to insert, delete and print datas from AVL Tree insertion in C Program

Text

Description automatically generated

1. Insertion

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

Text

Description automatically generated with medium confidence

1. Deletion

In this menu, the program will asked the value that the user want to delete from AVL tree. If the value is found in the tree, than it will be deleted otherwise the program gives message ‘data not found’. Do the deletion in (a).

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

1. Traversal

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

Text

Description automatically generated

1. Exit

Text

Description automatically generated