BINUS University

Academic Career:	Class Program:
*Undergraduate / ~~Master / Doctoral~~ *)*	*~~International~~/Regular/~~Smart Program~~/Global Class*)*
☐ Mid Exam ☐ Final Exam ☑ Short Term Exam ☐ Others Exam : _____	**Term : ~~Odd/Even~~/Short *)**
☑ **Kemanggisan** ☑ **Alam Sutera** ☑ **Bekasi** ☐ **Senayan** ☐ **Bandung** ☐ **Malang**	**Academic Year :** **2020 / 2021**

Faculty / Dept. : School of Computer Science	Deadline	Day / Date	:	Tuesday / Sept 14th , 2021
		Time	:	13:00 - 16:20 (200 Minutes)

Code - Course : COMP6048 – Data Structures	Class	:	All Classes
Lecturer : Team	Exam Type	:	Online

*) *Strikethrough the unnecessary items*

The penalty for CHEATING is DROP OUT!!!

Learning Outcomes:

LO 1 : Explain the concept of data structures and its usage in Computer Science
LO 2 : Illustrate any learned data structure and its usage in application
LO 3 : Apply data structures using C

EXAM INSTRUCTIONS

1. There are **2 parts** in this exam, **Essay** and **Programming Case**
2. For essay problem:
   a. You can **convert all the answers into 1 pdf file** and name the file using the following format: *nim.pdf*
3. For programming case problem:
   a. **The code that you submit has to be in .cpp file format** and name the file using the following format: *nim.cpp*
   b. For answering the question about **explanation in number 2**, **you can add your answer to the pdf file of your essay answer**.
4. All of your answers, **both essay (nim.pdf) and case (nim.cpp) have to be zipped** and submitted to the **htpps://exam.apps.binus.ac.id/**. The submission through other app will not be accepted for any reasons. (**Note: please zip both files using the following format:** *nim.zip*)
5. **The exam will be marked as 0 if any plagiarism is found**.
6. **The total duration of this exam is 200 minutes**, including the time for downloading the problem and uploading your answers. Please use the time provided wisely.

Verified by,

[Ajeng Wulandari] (D6422) and sent to Program on Aug 23, 2021

I. Essay (60%)

1. **[LO 1, LO 2, 12 points]** Using a stack, evaluate the following postfix expression:

5	12	2	1	+	3	1	-	*	/	3	+	*

In your simulation, you must show the content of stack and the operation performed on the stack at each step.
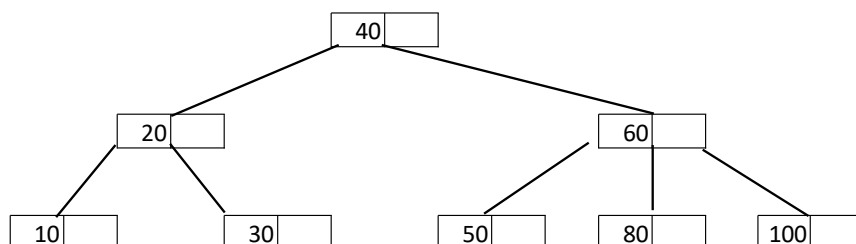
2. **[LO 1, LO 2, 8 points]** Mrs. Cook is selling her home-made cakes and cookies. Since there are a lot of type and varieties, she has to store her recipes somewhere such that it would be easy and quick to search for the recipes. You, as a friend of Mrs. Cook suggests her to use a hash table to store the recipes. The Recipe ID is a 5-digit code, and the size of the hash table is 80, where the address is from 00 to 79. Determine the address for the following ID, using linear probing when a collision occurs.
   - i. 00050
   - ii. 10021
   - iii. 20100
   - iv. 21141
   - v. 36079
   - vi. 40959

3. **[LO 1, LO 2, 12 points]** Consider the following Red Black Tree:
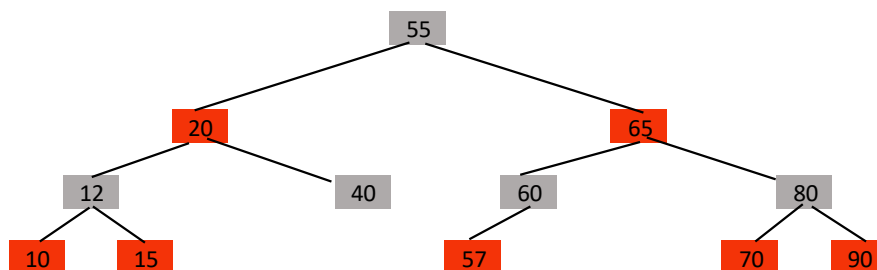
   - i. From the above tree, add the following numbers in sequence: 25, 30 and 27.
   - ii. From the resulting tree in (i), delete the following numbers in sequence: 30, 55

4. **[LO 1, LO 2, 8 points]** Consider the following B-Tree of order 3.


   - i. From the above tree, add the following numbers in sequence: 70, 15, 5.

    ii.   From the resulting tree in (i), delete the following numbers in sequence: 70, 40

5. **[LO 1, LO 2, 8 points]** Create a min-max heap from the following sequence of numbers: 20, 40, 10, 70, 5, 15, 3, 25. And then from the resulting tree delete minimum and delete maximum.

6. **[LO 1, LO 2, 12 points]** A car is almost out of petrol and must go to the nearest petrol station. There are several routes that the car can take, however because of limited petrol, the diver needs to find the shortest way from his current position to the petrol station. The map with distances can be shown as follows:
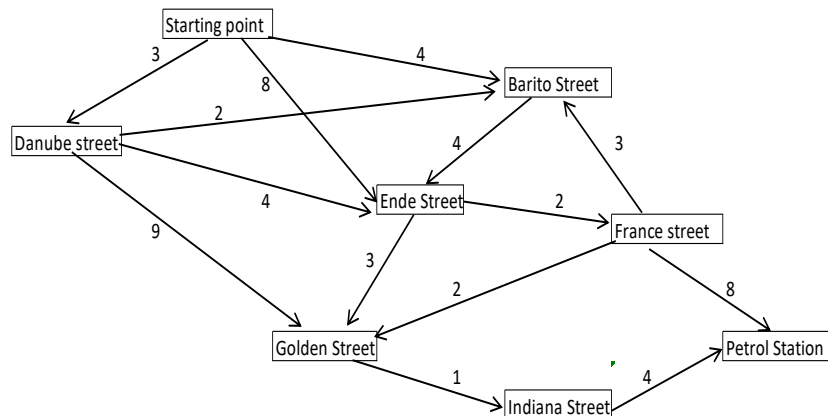
Help the driver to find the shortest route from the starting point to the petrol station using Dijkstra's algorithm, showing each step.

II.   Programming Case (40%)

1. **[LO 1, LO 2, LO 3, 15 points]** A mini mart needs to computerize the items that they are selling. The items in the mini mart consists of Item Codes, Description, Quantity and Price. Write a program using C programming to store the items in a single linked list, ordered based on the Item Code. The system should be menu driven, where the main menu is as follows:

```
*****MAIN MENU *****
1: Add Items
2: Display Items
3: Edit Items
4: Delete Items
5: EXIT

Enter your option : _
```

Option 1: Adding the items into the linked list. Note that when adding new items, you must check at which position that the items need to be inserted. The resulting list should always be in sorting order by Item number.

Option 2: Display all items in the list.

Option 3: Given an item code, find the items in the list. If found, then edit the data. If the item is not found, then write a message that the item doesn't exists.

Option 4: Delete one item from given item code, find the items in the list. If found, then delete the data. If the item is not found, then write a message that the item doesn't exists.

Option 5: Exit the program

2. **[LO 1, LO 2, LO 3, 25 points]** You wanted to perform a research to compare searching items using a linked list and using binary search tree. Using the same structure from question 1, write a menu driven program displaying the following menu:

```
***** COMPARE SEARCHING USING LINKED LIST AND BINARY SEARCH TREE *****
1: Add Items to the linked list and the BST
2: Display Items
3: Find Item
4: EXIT

Enter your option : _
```

Option 1: Ask user to enter a number and your program will enter the number in a linked list and the same number in a Binary Search Tree. You should use two functions, one function to add the number to the linked list (you may use the function that you have written for question 1), and another function to add the same number to a Binary Search Tree.

Option 2: Display the items in the list and display the items in the tree.

Option 3: In this option, 2 things need to be done. First, find the item in the list. While finding, add a counter to store the number of comparisons performed until the item is found, display the result on screen. Second, find the item in the Binary Search Tree. Similarly, add a counter to store the number of comparisons performed until the item is found, display the result on the screen.

Option 4: Exit the program

Question:
Please explain based on your result, which of the search (find) is better?

-- Good Luck --