**BACHELOR OF COMPUTER SCIENCE**

**SCHOOL OF COMPUTER SCIENCE**

**BINA NUSANTARA UNIVERSITY**

**MALANG**

**ASSESSMENT FORM**

**Course:** **COMP6360004 - Algorithm and Programming**

**Method of Assessment:** **Case Study**

**Semester/Academic Year : 1/2024-2025**

**Name of Lecturer : ………………………**

**Date : ………………………**

**Class : ………………………**

**Topic : Material Review II**

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

**Student Outcomes:**

**(SO 2) Mampu merancang solusi aplikasi piranti lunak berdasarkan analisis permasalahan yang dapat diselesaikan dengan pendekatan terstruktur dalam bidang informatika;**

***Able to design software application solutions based on problems analysis which can be solved with structured approach in informatics area*;**

**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-2 : apply syntax and functions in C language in problem solving**

**LO-3 : construct a program using C language in problem solving**

| **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 | LO 2 – LObj 2.2 – SO 2 | Ability to apply C syntax for problem solving | **50%** | 85% - 100% of C syntax is correctly applied | 75% - 84% of C syntax is correctly applied | 65% - 74% of C syntax is correctly applied | 0% - 64% of C syntax is correctly applied |  |  |
| 2 | LO 3 – LObj 2.2 – SO 2 | Ability to construct the algorithm into C program | **50%** | The C program is built 85% - 100% correctly | The C program is built 75% - 84% correctly | The C program is built 65-74% correctly | The C program is built 0-64% correctly |  |  |
|  | **Total Score:** ∑(Score x Weight) | | | | | | | |  |

Remarks:

**ASSESSMENT METHOD**

Instructions

1. This case study can be solved maximum in one team is 3 member, with duration of 2 week.
2. This case study consists of 2 questions.
3. The first question is a single case study problem, while the second question is a narrative case study   
   to **create 4 functions** with their respective goals. Your program should run correctly to get full score.
4. The second question should be combined into a single .c file. Give comments to explain which function does what.
5. All answers should be included into a single .zip file and submitted to Binusmaya.
6. Example only serves as an example. You may create the command line program as creative as you can.

**Note for Lecturers**:

1. This case study assignment will be held with duration of 2 week in review topic, or week 13.
2. The answer is manually checked by each lecturer (not by system).
3. The example only serves as an example to help students understand the assignment.   
   As long as the function works as intended and the program run correctly, you may give the full score for each problem.
4. Case study 1 **(LO 2 – L.Obj 2.2 – SO 2, 30%)**:

Given the array data as follows;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **5** |
| **7** | **2** | **4** | **3** | **9** | **8** |

1. Sort by ascending to the following data and simulate its sequence each round using Bubble sort.
2. Sort by ascending to the following data and simulate its sequence each round using Selection sort.
3. Sort by ascending to the following data and simulate its sequence each round using Insertion sort.
4. Simulate the searching process for key number of 3 using Binary search.
5. Simulate the searching process for key number of 9 using Interpolation search.
6. Case study 2 related to building the inventory management system **(LO 3 – L.Obj 2.2 – SO 2, 70%)**:

You are tasked to create a C program that manages an inventory system. This system will utilize three structs minimum to represent different entities.

Item’s struct : this struct will hold information about an individual item, such as its item’s name, price, and quantity.

Supplier’s struct : this struct will store details about the supplier of an item, including their name and phone number.

Inventory struct : this struct will items and suppliers, keeping track of the item’s ID, the Supplier’s ID, and the quantity of the item in stock.

Structure example (Note : You are allowed to modify this structure by adding additional type data):

|  |
| --- |
| struct Item **{**  int id**;**  char item\_name**[**50**];**  float price**;**  **};**  struct Supplier **{**  int id**;**  char name**[**50**];**  char phone\_number**[**20**];**  **};**  struct Inventory **{**  int item\_id**;**  int supplier\_id**;**  int quantity**;**  **};** |

**In addition to the data stored in the form of a binary file (**point 10 %**)**, the program should offer the following functionalities:

1. **Main Menu** (point 10%) : the programs can show the menu of : Insert Record, Show Record, Select Data, Delete Record, Update Record

INVENTORY SYSTEM

INVENTORY DATA MAIN MENU

What do you want to do?

1. Insert Record
2. Show Record
3. Select Data
4. Delete Record
5. Update Record

Your choice: 2

Number of rows: 3

Total Price: 24000

Total Stock Quantity: 56

------------------------------------------------------------------------------

ITEM\_ID SUPPLIER\_ID ITEM\_NAME PRICE QUANTITY SALES

------------------------------------------------------------------------------

A1 S1 BANANA 5.000 10 50.000

A2 S2 AVOCADO 7.000 6 42.000

A3 S1 STRAWBERRY 12.000 40 480.000

------------------------------------------------------------------------------

Grand Total 56 572.000

==============================================================================

1. **Insert Record** (point 10%) : add a new item to the inventory.
2. **Show Record** (point 10%) : display the entire inventory data.
3. **Select Data** (point 10%) : the programs can find the specific item’s.
4. **Delete Record** (point 10%) : remove an item from the inventory.
5. **Update Record** (point 10%) : modify the details of an existing item.