MODULE NAME:	MODULE CODE:
PROGRAMMING LOGIC AND DESIGN	PRLD5121/d/p

ASSESSMENT TYPE: ASSIGNMENT 1 (PAPER ONLY)

TOTAL MARK ALLOCATION: 100 MARKS

TOTAL HOURS: 10 HOURS

By submitting this assignment, you acknowledge that you have read and understood all the rules as per the terms in the registration contract, in particular the assignment and assessment rules in The IIE Assessment Strategy and Policy (IIE009), the intellectual integrity and plagiarism rules in the Intellectual Integrity Policy (IIE023), as well as any rules and regulations published in the student portal.

INSTRUCTIONS:

1. ***No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks. No more than 10% of the assignment may consist of direct quotes.***
2. ***Any assignment with a similarity index of more than 25% will be scrutinised for plagiarism. Please ensure that you submit your assignment through SafeAssign.***
3. ***Save a copy of your assignment before submitting it.***
4. *Assignments must be typed unless otherwise specified.*
5. *All work must be adequately and correctly referenced.*
6. *Begin each section on a new page.*
7. *This is an individual assignment.*

Referencing Rubric

Providing evidence based on valid and referenced academic sources is a fundamental educational principle and the cornerstone of high-quality academic work. Hence, The IIE considers it essential to develop the referencing skills of our students in our commitment to achieve high academic standards. Part of achieving these high standards is referencing in a way that is consistent, technically correct and congruent. This is not plagiarism, which is handled differently.

Poor quality formatting in your referencing will result in a penalty **of** according to the following guidelines **a maximum of ten percent being deducted from the overall percentage**. Please note, however, that **evidence of plagiarism in the form of copied or uncited work (not referenced), absent reference lists, or exceptionally poor referencing, may result in action being taken in accordance with The IIE's Intellectual Integrity Policy (0023)**.

Markers are required to provide feedback to students by indicating **(circling/underlining) the information that best describes the student's work.**

Minor technical referencing errors: 5% deduction from the overall percentage. – the student's work contains **five or more errors** listed in the minor errors column in the table below.

Major technical referencing errors: 10% deduction from the overall percentage. – the student's work contains **five or more errors** listed in the major errors column in the table below**.**

If both minor and major errors are indicated, **then 10% only** (and not 5% or 15%) is deducted from the overall percentage. The examples provided below are not exhaustive but are provided to illustrate the error.

Required: Technically correct referencing style	Minor errors in technical correctness of referencing style Deduct 5% from overall percentage. Example: if the response receives 70%, deduct 5%. The final mark is 65%.	Major errors in technical correctness of referencing style Deduct 10% from the overall percentage. Example: if the response receives 70%, deduct 10%. The final mark is 60%.
Consistency • The same referencing format has been used for all in-text references and in the bibliography/reference list.	Minor inconsistencies. • The referencing style is generally consistent, but there are one or two changes in the format of in-text referencing and/or in the bibliography. • For example, page numbers for direct quotes (in-text) have been provided for one source, but not in another instance. Two book chapters (bibliography) have been referenced in the bibliography in two different formats.	Major inconsistencies. • Poor and inconsistent referencing style used in-text and/or in the bibliography/ reference list. • Multiple formats for the same type of referencing have been used. • For example, the format for direct quotes (in-text) and/or book chapters (bibliography/ reference list) is different across multiple instances.
Technical correctness • Referencing format is technically correct throughout the submission. • The correct referencing format for the **discipline** has been used, i.e., either **APA,** OR **Harvard** OR **Law** • Position of the reference: a reference is directly associated with every concept or idea. • For example, quotation marks, page numbers, years, etc. are applied correctly, sources in the bibliography/reference list are correctly presented.	Generally, technically correct with some minor errors. • The correct referencing format has been consistently used, but there are one or two errors. • Concepts and ideas are typically referenced, but a reference is missing from one small section of the work. • Position of the references: references are only given at the beginning or end of every paragraph. • For example, the student has incorrectly presented direct quotes (in-text) and/or book chapters (bibliography/reference list).	Technically incorrect. • The referencing format is incorrect. • Concepts and ideas are typically referenced, but a reference is missing from small sections of the work. • Position of the references: references are only given at the beginning or end of large sections of work. • For example, incorrect author information is provided, no year of publication is provided, quotation marks and/or page numbers for direct quotes missing, page numbers are provided for paraphrased material, the incorrect punctuation is used (in-text); the bibliography/reference list is not in alphabetical order, the incorrect format for a book chapter/journal article is used, information is missing e.g. no place of publication had been provided (bibliography); repeated sources on the reference list.
Congruence between in-text referencing and bibliography/ reference list • All sources are accurately reflected and are all accurately included in the bibliography/ reference list.	Generally, congruence between the in-text referencing and the bibliography/ reference list with one or two errors. • There is largely a match between the sources presented in-text and the bibliography. • For example, a source appears in the text, but not in the bibliography/ reference list or vice versa.	A lack of congruence between the in-text referencing and the bibliography. • No relationship/several incongruencies between the in-text referencing and the bibliography/reference list. • For example, sources are included in-text, but not in the bibliography and vice versa, a link, rather than the actual reference is provided in the bibliography.
In summary: the recording of references is accurate and complete.	In summary, at least **80%** of the sources are correctly reflected and included in a reference list.	In summary, at least **60%** of the sources are incorrectly reflected and/or not included in reference list.

Overall Feedback about the consistency, technical correctness and congruence between in-text referencing and bibliography:

At the end of this assignment, you should be able to:

LU1 - LO4: Design pseudocode to solve programming problems.

LU1 - LO6: Create mathematical expressions;

LU1 -LO7: Design a programme by breaking the problem up into modules;

LU2 - LO7: Write pseudocode using counted loops and conditional loops;

LU2 - LO8: Include nested loops in your programming solution.

LU3 - LO3: Write pseudocode that searches for a specific value in an array;

LU3 - LO5: Explain why data should be sorted;

LU3 - LO6: Write pseudocode to sort an array using a bubble sort.

LU4 - LO4: Write pseudocode with methods that require no parameters;

LU4 - LO5: Write pseudocode with methods that require parameters;

LU4 - LO6: Write pseudocode with methods that return data;

LU4 - LO7: Describe how to pass an array to a method;

Question 1 **(Marks: 45)**

Write the pseudocode to solve the problem below. Marks will be awarded for meaningful comments.

The HR manager has requested that the IT department create a salary report for the CEO. This task has been assigned to you. The report will be for three departments and five employees spread across those departments. The HR manager has asked for you to make allowances for the following data, which is required for the report:

- Employee's name (five employees)
- Department 1, 2, or 3 (each employee belongs to a department)
- The monthly salary for each employee.

Declare and initialise three parallel arrays using the values shown in the arrays below. Do not ask the user to input values. The arrays are already sorted on the values in the department array.

Employee name

Bongi	Tyrone	Priya	Joseph	Jason
0	1	2	3	4

Department

1	1	2	2	3
0	1	2	3	4

Monthly Salary

33000	25000	85000	12000	50000
0	1	2	3	4

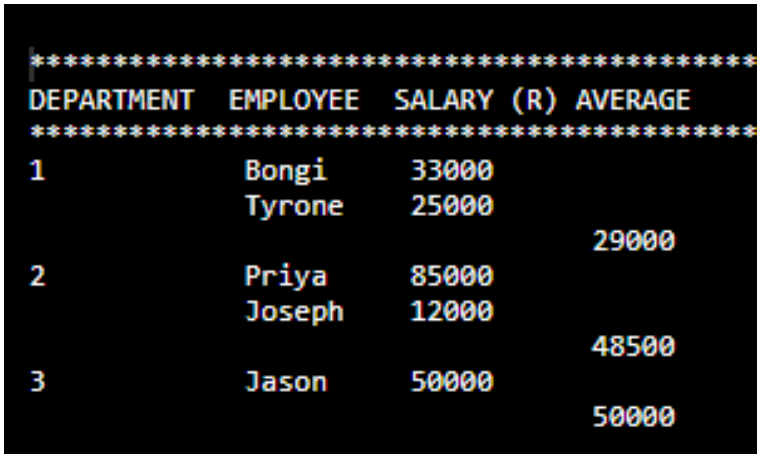
Your solution must do the following:

1. declare and initialise variables and the three parallel arrays.
2. input values into the three parallel arrays. You do not need to consider error handling on the input; assume that the user never makes a mistake (unlikely, but we are not concerned about it for this assignment).

3. calculate the average salary per department.
4. display the report.

Calculate the average monthly salary per department and print the report. The average monthly salary must be calculated for each department. The department number must not be repeated for each employee. The department number is only displayed when there is a new department.

The report must be in this format:



Follow these marking guidelines:

COMPONENT TO BE ASSESSED	MARKS
Declare and initialise variables	3
Declare and populate the 3 arrays	6
Display the headings in the correct format	3
Display the initial department number	2
Increment subscript value	2
While loop	3
Determine if it is a new department or not (if…else…)	6
If new department the display department and employee's details	6
If not a new department the only display the employee details.	6
Calculate the average salary per department	4
Display the average salary for each department	2
Meaningful Comments	2
TOTAL	**45**

Question 2 **(Marks: 30)**

Write a report that covers the following topics:

1. Explain a Bubble Sort and two other sorting algorithms.
2. Explain which of those sorting algorithms you prefer, and why.
3. Include diagrams to explain the logic for each of the three sorting algorithms.

There are multiple sorting algorithms you can choose from, for example: insertion sort, merge sort, selection sort, etc. Your report must not exceed five typed pages in length (including diagrams). Include at least one external source and reference it.

Use this marking guide to identify what should be included in the report.

COMPONENT TO BE ASSESSED	MARKS
Definition of sorting	3
Explanation of a Bubble Sort	5
Explanation of sorting algorithm 1	5
Explanation of sorting algorithm 2	5
Which sorting technique is preferred? Why is that sorting technique preferred?	3
Diagram - Bubble Sort	3
Diagram - Sorting algorithm 1	3
Diagram - Sorting algorithm 2	3
TOTAL	**30**

Question 3 (Marks: 25)

Consider the following problem statement:

You are tasked with writing the pseudocode for a programme that performs various operations on an array of integers. The size of the array is five. The user must enter the values to be stored in the array (use a loop). Enter the values in ascending order (already sorted).

Implement the following operations in each of the four methods:

- findMax: Takes the integer array as input and displays the highest value in the array.
- findMin: Takes the integer array as input and displays the lowest value in the array.
- calculateAverage: Takes the integer array as input and displays the average value of the elements in the array.
- reverseArray: Takes the integer array as input and displays the elements reversed.

Your pseudocode must include the above four methods and demonstrate their functionality by populating the array and then calling each individual method and displaying the highest value, lowest value, average value, and reversed array in the relevant method. You must include meaningful comments.

Follow these marking guidelines:

COMPONENT TO BE ASSESSED	MARKS
Declarations	1
Input 5 values into the array using a for-loop	2
Call the methods (1 mark per method)	4
findMax method	4
findMin method	4
calculateAverage method	5
reverseArray method	5
TOTAL	**25**

[TOTAL MARKS: 100]

END OF PAPER