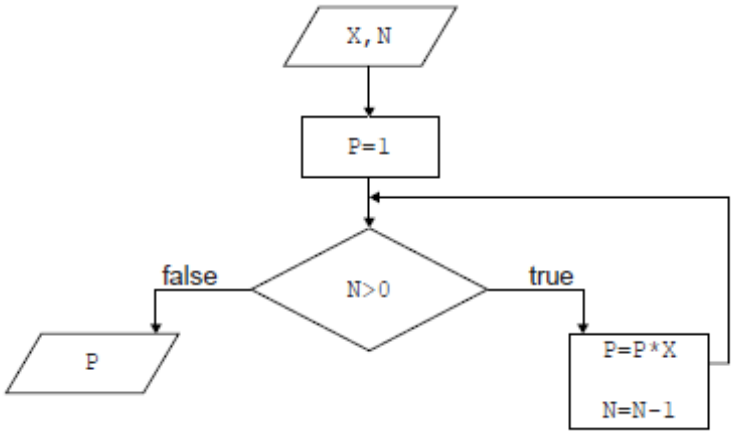
SL Unit 4 – Problem Solving

Quiz 4

Question 1			
Objectives:	4.2.4	Exam Reference:	Nov-16 6

1. Consider the following algorithm that inputs *X* and *N*, and outputs *P*.



(a) Determine how many times multiplication is performed when this algorithm is executed. [1]

N;

(b) Construct a trace table for the algorithm when X=2 and N=4. Use the following headings in your table.

X	N	P	N>0	output

[4]

*Award **[1]** for each of four correct columns (with headings N, P, N>0 and output*

X	N	P	N>0	output
2	4	1	true	
	3	2	true	
	2	4	true	
	1	8	true	

(c) State the purpose of this algorithm. [1]

Calculates X^N;
Note: *DO NOT accept vague answers that may suggest the understanding of* N^X *or use incorrect terminology*

Question 2			
Objectives:	4.2.1, 4.2.7	Exam Reference:	May-15 10

2. (a) Identify **two** differences and **two** similarities between a bubble sort and a selection sort when sorting an array of 10 elements. [4]

*Award **[2 marks max]** for the similarities and **[2 marks max]** for the differences.*
Both use nested loops;
Each time reducing the inner loop;

Bubble sort swaps adjacent items each time it goes through the list;
Selection sort finds the next smallest each time it goes through the list;
Bubble sort can exit early if already sorted;

A cycling tour lasts for 15 days. The total time for each competitor is recorded in a one-dimensional array, *TIMES[].* After each day's race, the array entry for each competitor is increased by their time for that day.

There are 150 competitors and the 10 fastest times are transferred to the array *FASTEST[]* and displayed on a screen each day.

(b) Explain why a selection sort would be more efficient than a bubble sort in this case. [2]

*Award **[1 mark]** for reference to the size of the list and **[1 mark]** for stating why the selection sort is faster.*
It is possible that selection sort will only need 10 passes of the outer loop to find 10 fastest times;
But bubble sort will need to complete the procedure for the entire list every time;

OR

List is long;
Swapping takes longer than selecting;

(c) Construct an algorithm to transfer the 10 fastest times from the array *TIMES[]* to the array *FASTEST[].* Assume that the array *TIMES[]* is not sorted. [6]

*Award marks as follows up to **[6 marks max]**.*
*Award **[1 mark]** for setting all variables at start.*
*Award **[1 mark]** for correct double looping through `TIMES`.*
*Award **[1 mark]** for correct selection of `MIN`.*
*Award **[1 mark]** for transfer to `FASTEST`.*
*Award **[1 mark]** for successful swapping selected value.*
*Award **[1 mark]** for resetting `MIN` to value transferred.*
*Award **[1 mark]** for incrementing array index in `FASTEST`.*
*Award **[1 mark]** for transfer of exactly 10 elements.*

For example:

```
MIN = 0
TRANSFER = 0
loop while TRANSFER < 10
     MIN = 250000 //larger than first 10 fastest
     COUNT = TRANSFER
     loop while COUNT < 150
          if TIMES [COUNT] < MIN
               MIN = TIMES[COUNT]
               K = COUNT
          end if
          COUNT = COUNT + 1
     end loop
     FASTEST[TRANSFER] = TIMES[K]
     TEMP = TIMES[TRANSFER]
     TIMES[TRANSFER] = TIMES[K]
     TIMES[K] = TEMP
     K = K + 1
     TRANSFER = TRANSFER + 1
end loop
```

***Note:** Accept any reasonable value set as `MIN` – including `TIMES[0]` provided that this is not replaced by `MIN` after the first loop.*

Alternatively the array can be sorted and then transferred in which case award marks as follows:
*Award **[1 mark]** for creating the array `FASTEST`.*
*Award **[1 mark]** for correct double looping through `TIMES`.*
*Award **[1 mark]** for comparing adjacent values.*
*Award **[1 mark]** for correct swap if second value is lower.*
*Award **[1 mark]** for looping through `FASTEST`.*
*Award **[1 mark]** for transferring first `TIMES` to `FASTEST`.*

The race organizers need to display the names of the 10 fastest competitors, as well as their times, on the screen. There is another array, *NAMES[],* which contains the names of all competitors in the same order as their times in *TIMES[]* (for example, *NAMES[5]* and *TIMES[5]* are the name and time of the same competitor).

(d) Compare the use of two arrays, to hold the competitor's times and names, with the use of objects. [3]

The problem with parallel arrays is the sorting/indexing/maintaining relationship;
An object would contain at least a name and a time (accept other descriptions of object);

Would only need to sort the array of objects / only one list to be sorted;

Question 3

Objectives:	4.1.16	Exam Reference:	May-15 7

Six students are planning their group 4 project, which is due in two days. They have to produce a scientific report and give an animated computer presentation based on their analysis of water samples. These water samples are to be taken from four local lakes.

(a) Based on this information, identify **four** tasks that should be done by the students, listing the tasks in the order that they could be completed. [2]

*Award **[1 mark]** for the correct list and **[1 mark]** for the correct order. The last two can be in any order. Accept similar descriptions of individual tasks.*
Collect samples
Analyse samples
Write report
Prepare presentation

(b) Outline how **two** of the tasks identified in part (a) could be completed concurrently. [2]

*Award **[1 mark]** for the correct tasks and **[1 mark]** for outlining how they can be done concurrently*.

"Write report" and "prepare presentation" can be done at the same time as they can be performed by different students (using the same data).

(c) Draw a Gantt chart to show the tasks from part (a), indicating the concurrency outlined in part (b). You do not need to include the timings for the tasks. [2]

*Award **[1 mark]** for a correctly labelled chart illustrating the order of tasks from part (a) and **[1 mark]** for showing the concurrency from part (b).*
For example:

Task	Phase 1	Phase 2	Phase 3
Collect samples	▬▬▬		
Analyse samples		▬▬▬	
Write report			▬▬▬
Prepare presentation			▬▬▬

Question 4

Objectives:	4.3.4	Exam Reference:	May-17 10

Other than the use of different keywords, outline two ways in which two higher level languages might differ from one another. [4]

*Award **[1]** for identifying a way in which a higher-level language may differ from another and **[1]** for developing that difference, up to **[2 max]**.*

Mark as 2 and 2

Method of translation;
Whether by compiler or interpreter (or both);

Loosely/strongly typed;
Refers to whether data types are specified;

Different programming paradigms;
Procedural or object oriented etc.;

Purpose of the language;
Specific (*eg* scientific/AI) or general;

Compatibility with different environments;
Java with virtual machine can run on all O/S / some languages are O/S specific;

Syntax differences;
Structure of statements *etc*;

Note: *Accept other reasonable answers*