May 2016 SL P1

Section A

1. Outline **one** problem of maintaining legacy systems. [2]

*Award up to **[2 max]**.*
*Award **[1]** for evidence that the candidate knows what is meant by "maintaining legacy systems".*
*Award **[1]** for any appropriate problem outlined.*

Example:
Maintaining previous/outdated computer system, which uses old technology and
old application programs ;
That are hard to understand/expensive to change/evolve **because** programs might be disorganized/
documentation might be missing/incomplete/unreliable;
Compatibility issues (typically refer to old programming languages/old database technology);
Maintaining but not updating the old system can lead to compatibility / security issues;
It may be difficult to recruit staff/programmers familiar with old languages/operating systems;
Database contains inconsistencies/redundancies (eg information systems and no DBMS);
Usually pre-internet, needs interfaces;
Typically large and complex systems/mainframes;

2. Explain what is meant by user acceptance testing. [2]

A system or a partially functional prototype;
Given to users to test for functionality or to gain feedback on functions or the user interface;

3. Discuss one advantage and one disadvantage of printed material, when compared to online support, as a method to provide user documentation. [4]

*Award **[1]** for **one** advantage and **[1]** for **one** disadvantage, and **[1]** for expanding on each of them, up to **[4 max]**.*
Examples of advantages:
• Portability;
Printed material is more easily transportable and can be moved around (*eg* for a scanner/printer);
• Extent of material;
Books/technical instructions for installation may have more details, and be more useful to provide deeper explanations;
• Availability;
It is always available (no power cut problems);
Examples of disadvantages:
• Readability;
Font size online can be (easily) magnified;
• Trouble-shooting/cross-reference/usability;
Online is faster and usually has links to other pages whereas paper is a thick manual;
• Aging;
Online is more frequently updated than paper manuals;
• Environmental;
Waste of paper versus energy consumption;

4. Outline the use of a failover system. [2]

*Award up to **[2 max]**.*
A failover system is a standby/redundant system;
Which is used to eliminate/reduce the impact on users/owners;
By automatically taking over if the primary system suddenly becomes unavailable;

5. Describe the function of the control unit (CU) in the central processing unit (CPU). [2]

*Award up to **[2 max]**.*
Obtains the data/instructions from the memory;
Interprets/decodes them into commands/steps/signals;
Controls transfer of data and instructions among other units of a CPU (for example, command to ALU for execution);
Manages/coordinates all the units of the computer;
etc.

6. Describe how the cache memory can speed up the functioning of a processor. [2]

*Award up to **[2 max]**.*
Cache memory is closer to CPU/faster to access than main memory/incorporated on the chip;
By holding recently/frequently used data and instructions in cache;
Execution of program/fetching instructions and data is faster;

7. Outline one feature of the operating system that needs to be considered when running a game application. [2]

Examples:
Memory management;
Game applications use a lot of memory and require constant refreshing;
Processor loading of OS functions (efficiency);
Graphics handling of OS (as distinct from graphics card),
GUI;
O/S needs to handle input from appropriate devices;

8. Construct the truth table for the following expression.

```
A xor (B or C)
```

[3]

Award [3] marks for all 8 correct rows in the truth table.
Award [2] marks if only 6/7 correct rows in the truth table.
Award [1] mark if only 5 correct rows in the truth table.

A	B	C	A XOR (B OR C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

9. In an 8-bit register, state the binary representation of the hexadecimal number *3B*. [2]

*Award **[1]** for correct 111011.*
*Award **[1]** for using two leading zeroes for the 8-bit register.*

00111011;

10. Trace the following fragment, for N=139 and L=3, by copying and completing the trace table given below.

```
D = N div L
Z = 1
B = false

loop while Z<L
  D = D div L
  Z = Z+1
  B = NOT B
end loop
if (D ≠ 0 AND B) then
  output(D, B)
else
  output(Z, NOT B)
end if
```

D	Z	B	Z<L ?	output
...	...	...	...	...

[4]

*Award **[1]** for correct output.*
*Award **[1]** for each of the **three** correct iterations (three rows).*
***Note:** Accept 129/3, 139/9 and 139/27* D.

D	Z	B	Z<L ?	output
46	1	false	true	
15	2	true	true	
5	3	false	false	
				(3 , true)

Section B

11. An examination office of a university must securely store students' examination papers and their grades. The office keeps the documentation of past students for two years. After two years the office only stores the student grades. All documentation of current students is frequently accessed for other operations and the volume of the data increases quickly.

    To better support its operations, the office is creating a new system to provide this storage.

    (a) Identify **two** aspects of the data that need to be taken into account during the planning of the new system. [2]

*Award up to **[2 max]** for **two** aspects:*
The type of access needed;
For example read only/read write/online or offline;

Access rights;
For example, data available only for administrators / different permissions for students;

Frequency of access;
Some data (of non-current students) are not frequently accessed and can be archived;
Other data (of current students) are frequently used, subject to a variety of operations;

Quantity/size of the data;
For example should not exceed storage capacity of the new system;

Type/nature/format;
For example incompatibility issues;

    (b) Describe how direct observations on the current system may provide information to help propose a suitable new system. [3]

*Award up to **[3 max]**.*
Quick/first hand/realistic information on data/software/hardware/users/procedures in the current system;
Help better understand positive and negative features of the current system (for example problems in accessing or validating data/user errors/security issues, etc.);
Which can be used when specifying requirements of the new system (keep/improve positive and change negative features);

A prototype of the new system is created to present to the examination office.

(c) Describe the purpose of this prototype. [3]

*Award up to **[3 max]**.*
Prototype is used to ensure all essential functions/operations of the system are present/meets the needs of the users;
Prototype is used to speed up development process;
Positive user's feedback helps in refining the acceptable prototype in order to develop the complete system/product;
Or else a further prototype should be created in order to develop the satisfactory system/product;

The examination office needs to upgrade the computing resources for their operations, and this will require data migration.

(d) Discuss **two** possible problems that may occur during data migration. [4]

*Award **[1]** for identifying a problem and **[1]** for a discussion, for **two problems**, up to **[4 max]**.*

Data loss/data corruption;
When moving data, from one storage device to another (via network/ cables or transferred by people), data could be corrupted/lost and not useful anymore;

Incompatibility of data formats;
Necessary to translate from one format to another, to be able to use the data in the new system which causes delays/performance issues in business/office operation;

(e) Outline **one** economic aspect that the examination office needs to take into account to support parallel running. [3]

*Award up to **[3 max]**.*

Example answer 1
Two systems are running simultaneously so that operations are not disrupted;
This is a costly operation;
Because both systems and all their resources should be maintained / More staff should be hired;

Example answer 2
Safe way of validating the new system;
Running two systems could be cheaper;
Than losing all data in case of failure;

12. A college has a high-speed network. The network is accessible to all students and staff through their personal accounts.

The network may be accessed by using desktop computers available in the college. When in the college, users can also use personal laptops to connect wirelessly or dock with an Ethernet cable. When not in the college, users can connect via a virtual private network (VPN) over the internet.

(a) In the given context, distinguish between Ethernet and wireless in terms of **reliability** of transmission. [4]

*Award **[2]** for an explanation of the reliability of wireless.*
*Award **[2]** for an explanation of the reliability of Ethernet*
*Award up to **[4 max].***
***Note:** Do not accept answers relating to security.*

WIRELESS
The reliability of wireless depends
on the strength of the wireless signal/distance from router;
on the topology/shape of the surroundings;
on interference/number of simultaneous connections on an access point;

ETHERNET
Ethernet is more reliable as the strength of the signal is independent from the distance from the router (within the college);
There is no issue with the topology/shape of the surrounding, as long as the user has a connection;
connection depends on condition of cables – no loose or broken cable connections;

(b) Describe **two** features of a VPN that make it secure. [4]

*Award **[1]** for identifying the feature, up to **two** features.*
*Award **[1]** for an expansion of **two** features.*
*Award up to **[4 max]**.*

Authentication;
Nobody outside the VPN should be able to affect the security property of the VPN (it must be impossible for the attacker to weaken/change encryption);

Encryption;
Data intercepted will not be readable;

Tunnelling software;
Security properties of each tunnel should be agreed by the administrators of the two endpoints of the tunnel;

Multiple exit nodes;
Makes it hard to distinguish where the data was generated thus more secure (less prone to phishing);

(c) State **one** technology that is necessary for a VPN. [1]

*Award **[1 max]**.*
SSL 3.0 (Secure socket layer 3);
TLS (with encryption) (Transport Layer Security);
IPsec with encryption;

***Note:** Do not accept "internet connection".*

The college is devising a policy for the use of its IT resources and services. They are considering prohibiting the use of external services such as cloud storage and blogs.

(d) In relation to the specific activities that may be carried out by students, discuss **two** advantages and **two** disadvantages of the use of external services. [6]

*Award **[1]** for **each** advantage, up to **[2 max]**, **[1]** for **each** disadvantage, up to **[2 max]**, and **[2]** for a discussion in relation to an example.*
Answers may include (and are not limited to) the following elements:

Advantages:
May offer more recent technology than the college itself;
The college cannot replicate the social aspects of diffused discussion/social networking;
Registration/creation/access of account is usually easy/cheap/free;
Allows interaction with others/collaborative studies/exchange of materials/opinions;
It means students can access and coordinate data and assignments on any device and from any location;
***Note:** Do not accept "can reach work when not at college" on its own, as there is a VPN for this purpose.*

Disadvantages:
One might post things they would regret later on;
The content is potentially available across the world;
The content is available for a long time;
The content submitted to external services might not be fully in line with the internal policy, even if posting it is allowed (offensive content);
Photos/documents may be hacked by third parties;

For example:
Posting offensive comments;
Ownership / confidentiality / security of data;
Takedown policy from external providers (if the posted content is offensive, the service provider may take a while to take it down);

13. A local charity organizes a half-marathon to raise money. The rules to participate in the half-marathon are as follows:

- The organizers limit the total number of participants to 450
- Participants belong to a team and each team must have at least three and at most five
- participants
- Each participant registers for the event independently from the other members of their
- team, and they all declare their team name when registering
- For scoring, the team's final time is the sum of the times of its three fastest participants. Participants that do not cross the finishing line within 2 hours after the start, are assigned a default time of 1000 minutes. The **winning team** is the team with the smallest sum total.

During registration, an array, *PARTICIPANTS*, with 450 positions is used to hold the abbreviated team names that are declared by each participant. Simultaneously, a collection *TNAMES* is generated: any new team name that is declared is added to the collection.

(a) State the minimum size of *TNAMES* to ensure the names of all potential teams can be stored. [1]

150 (= 450/3);

Part of the array *PARTICIPANTS* is shown below, where, for example, the first participant declared that they are part of team *TK*. The initial part of the collection *TNAMES* is also shown, with arrows indicating the direction of growth.

PARTICIPANTS

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	...
TK	W	AC	TK	W	TK	AC	W	TK	TK	AC	QA	AC	W	AC	...

TNAMES

Both *PARTICIPANTS* and *TNAMES* are used to construct the array, *TEAM*, that groups all participants who belong to the same team. Part of the array *TEAM* is shown below.

TEAM

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	...
3	4	6	5	7	8	10	13	9	0	12	73	14	15	2	...

In *TEAM*, each element is related to one other index in the array, shown by the arrows on the above diagram. This relation will eventually form a closed path (for this example 0, 3, 5, 8, 9 and back to 0). The relation reflects the information in *PARTICIPANTS*, by grouping people who declared the same team name during registration.

Hence, participants 0, 3, 5, 8 and 9 are on the same team and, from *PARTICIPANTS*, that team is *TK*.

(b) Identify the position in *PARTICIPANTS* of the second participant that registered for team *QA*. [1]

73;
Note: *Accept* `PARTICIPANTS[73]`. *Do not accept* `TEAM[11]` *because the question asks specifically for the array* `PARTICIPANTS`.

Part of the algorithm that generates the *TEAM* array is shown below, in pseudocode.

```
//Input PARTICIPANTS array, TNAMES collection
TEAM    // array with 450 positions, initialized to '999'
CURRENT // variable to store current name of team;
T, P    // variables to store the indexes of TEAM and PARTICIPANTS,
        // respectively;
MINP    // stores the first index P of members of the CURRENT team;

TNAMES.resetNext()
loop while TNAMES.hasNext()
       CURRENT = TNAME.getNext()
       T = 0; P = 0; MINP = 0     // variables' initialization
       //*
       //* Code to be completed in part (c)(i)
       //*
       //* Code to be completed in part (c)(ii)
       //*
end loop
output TEAM
```

(c) In order to complete this code, and return the correct *TEAM* array,

(i) construct pseudocode to find *MINP*, the first index in *PARTICIPANTS* of the *CURRENT* team, and use it to start the construction of *TEAM* [3]

*Award marks as follows up to **[3 max]**.*
*Award **[1]** for looping through PARTICIPANT.*
*Award **[1]** for checking PARTICIPANT [P] = CURRENT.*
*Award **[1]** for exit when MINP found – accept 'break'.*
*Award **[1]** for storing the index in MINP.*

Example 1:

```
loop while PARTICIPANTS[P] ≠ CURRENT AND P<450
   P = P+1
endloop
MINP = P
T = MINP
```

Example 2:

```
boolean FOUND = FALSE
loop while FOUND = FALSE and P < 450
   if PARTICIPANTS[P] = CURRENT then
      FOUND = TRUE
      MINP = P
   else
       P = P + 1
   end if
end loop
```

Example 3:

```
boolean FOUND = FALSE
loop P from 0 to 450-1
   if PARTICIPANTS[P] = CURRENT then
      FOUND = TRUE
      MINP = P
      BREAK
   end if
end loop
```

(ii) construct pseudocode to find the other participants belonging to the *CURRENT* team, implementing the idea of the closed paths in the *TEAM* array. [4]

*Award marks as follows up to **[4 max]**.*
*Award **[1]** for a correct loop from `P` to 449 (either incremented in (c)(i) or stated here).*
*Award **[1]** for starting at `T=MINP`*
*Award **[1]** for updating `T` index.*
*Award **[1]** for correct value of `P` in `TEAM[T]`.*
*Award **[1]** for closing the path by assigning `MINP`.*

Example 1:

```
T = MINP // can be here or in c(i)
loop while P<450
   if PARTICIPANTS[P] = CURRENT then
      TEAM[T] = P
      T = P
   end if
   P = P+1
end loop
TEAM[T] = MINP
```

Example 2:

```
loop while P<450
   if PARTICIPANTS[P] ≠ CURRENT then
      P = P+1
   else
      TEAM[T] = P
      T = P
      P = P+1
   end if
end loop
TEAM[T] = MINP
```

As part of the program to determine the winning team, an array, *TIMING*, is maintained in parallel to *PARTICIPANTS*. For example, *TIMING[5]* and *PARTICIPANTS[5]* relate to the same participant.

TIMING is initialized to zero before the race starts, and updated with the finishing times for each participant. The algorithm *sum3best* is able to output the sum of the three fastest times from any group of times that are passed to the algorithm.

(d) Describe the steps of an algorithm that will find the **winning team**, as defined by the marathon rules on page 6. Clearly mention the use of existing or of new data structures. [6]

*Award up to **[6 max]** for covering the following points.*

Describe and use new data structures/variables;
Loop through `TNAMES` for each team;
Retrieve/access `TIMINGS` for the times of each member from `TEAM`;
(Store in array or list)
Pass team member times to sum3best;
Store result of `sum3best` for team in array or variable;
Identify winning team;

Example 1

```
Create array TEAMTIMES with same length as TNAMES
Create array TEMPTIMES with length 5 (max number of participants per team)
For each team in TNAMES
    For each team member in TEAM find the corresponding time in TIMING
        Insert the time into the next element of TEMPTIMES (assumption: times over 120 min
        have already been assigned a time of 1000)
    End loop
    Pass TEMPTIMES to function sum3best
    Insert result of sum3best into TEAMTIMES at same position as current element of TNAMES
End loop
Find smallest value in TEAMTIMES
Set TEAM = TNAMES[index of smallest value in TEAMTIMES]
```

Example 2

```
Create array TTIMES max 5 to hold times in one team
Create FASTEST to hold time of winning team
Create BEST to hold name of winning team
Set FASTEST = 5000 (default time for 5 team members or something)
For each team in TNAMES
    use PARTICIPANTS and TEAM to find index for each member in TIMINGS
    store in TTIMES array of max 5
    Pass array TTIMES to sum3best and store in SUM
    If SUM < FASTEST set FASTEST to SUM and set BEST to TNAME
End loop
winning team = BEST
```