May 2016 HL 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Maintaining previous/outdated computer system, which uses old technology and old application programs ;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. Outline the use of a failover system. [2]

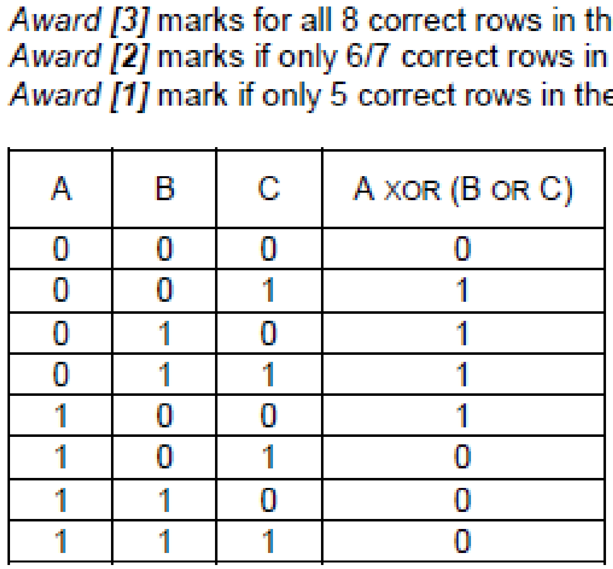
*Award up to **[2 max]**.*
A failover system is a standby/redundant system;
Which is used to eliminate/reduce the impact on users/owners;
By automatically taking over if the primary system suddenly becomes unavailable;

3. 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.

4. 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

5. 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;

6. The following list of numbers needs to be put into ascending order.

`9, 11, 3, 4, 5, 7, 1, 2`

State the list that would be obtained after **two** iterations of a bubble sort. [1]

3, 4, 5, 7, 1, 2, 9, 11; **[1]**
Note: *Some candidates might apply the bubble sort to the list, from right-to-left. The answer in this case would be* 1, 2, 9, 11, 3, 4, 5, 7, and this should receive full credit.

7. Outline why a virtual machine is an example of abstraction that is particularly useful when testing software on different platforms. [3]

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

Virtual machines provide a software emulation/virtualization of other operating systems;
While hiding/abstracting the physical environment/OS/execution environment of the host machine;
Virtual machines do not directly correspond to any real platform so software may be tested/executed on any platform (portability);
This is more practical/economical than having to test software on several computers with different OS;
The software to be tested is compiled into the language of the virtual machine, and this is interpreted into the language of the host machine;

8. Explain, in terms of resources, how Voice over IP provides a collaborative working environment for a company with multiple locations nationwide. [3]

Award ***[2]*** *for description of VoIP.*
Award ***[1]*** *for an explanation related to one (or more than one resource).*
Note: *Award* ***[1]*** *only if resources are mentioned without any explanation.*

Example:
Audio (and video) stream transmitted over internet;
Broadband internet connection needed;
That can be integrated in an office desktop computer;
with collaboration desk endpoints/webcams/desktop computer/widgets/
touch screen/specific applications/IP phones;

Examples:

Bandwidth:
Available Bandwidth / A trade-off between bandwidth and quality of audio-video streaming;
Low bandwidth makes streaming disrupted/discontinuous and impacts on the quality of the collaborative environment;
Bandwidth also depends on how many people in the office will use VoIP (and possibly the network in general) and their simultaneous operations (uploading/downloading);

Router:
Router should be powerful/good enough to support all operations related to collaborative work;
Some specific routers provide quality of service for voice transmission over other functions, for example WiFi/VPN connections;
Normal routers do not distinguish the internet traffic, and the quality of audio-video streaming may decrease;

Network:
Quality of internet varies, for example delays/congestion are cause of losses/echoes/scattered images;

Having two separated networks for VoIP and other computer operations may help in some cases (depending on the dimensions and operations of the office), so that the router can handle them separately;
For example somebody talking over the IP is not hindered by somebody else in the same office downloading from the internet;

Power:
In case of power loss the telephones will also not work;
A backup system/second system just for the telephones is necessary;

9. Explain **two** functions that an operating system needs to perform in relation to multitasking. [4]

*Award **[2]** for each of the two functions, up to **[4 max]**.*

Memory management;
OS allows more than one program/process to share the memory;
By allocating separate memory to each program;
Provides memory isolation for each of processes;
The system may begin to run out of shared storage (as many programs are running) so OS moves pages to disk/paging;
etc.

Processor Management;
To allow (the appearance of) more than one program running at the same time;
By the allocation of time slices;
Decides which process runs at a certain point in time;
Arrange the execution of applications so that you believe that there are several things happening at once (scheduler);
Prioritizes tasks by importance (interrupts);
etc.

10. Discuss **one** ethical consideration of using CCTV in a workplace. [3]

*Award **[1]** for a rationale, **[1]** for an example of misuse, and **[1]** for a way to prevent it, up to **[3 max]**.*

The rationale:
The right of the company to protect their premises/assets/IP shall not invade privacy rights of the employee;

Therefore precise guidelines must be set to prevent misuse of technology from the employer and ensure rights to the employee;

Examples of Misuse:
CCTV might not be used for surveillance only, but also to monitor employees;
It is a threat to personal privacy through intimidation/harassment/checking absentees/screening with consequences on quality of life/wellbeing/employment;

For example: monitoring employee’s activities/look for negative behaviour/gestures / focusing on some groups of employees (women/LGBT/ethnic or faith groups, *etc*);

For example: different cultures may interpret gestures in different ways, and this might be used to reprimand an employee;

Moving equipment (also a USB stick with data) from one building to another is monitored, but the footage can be interpreted as theft;

Prevent Misuse:
A secondary system for labelling equipment should be used (tagging items), to avoid misinterpretations, and discharge incorrect incriminations (but it is more subtle with IP);

Problem of where the footage is stored (company premises), for how long, and who has access to the information, for what use, shall also be addressed;

Surveillance staff (*ie* those who have access to CCTV footage) shall not be employees of the company or there could be internal conflicting situations (*ie* an employee controlling another employee);

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]**.*

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 car park has two barriers. One barrier is at the entrance, where tickets are **issued**, and one barrier is at the exit, where paid tickets are **checked** when cars leave. A display at the entrance, showing the current availability of spaces in the car park, is updated as tickets are **issued** and **checked**.

The actions of issuing, paying and checking a ticket operate on the collection of objects, *TICKETS*, that is organized as a linked list. Each object holds the following information:

Nr: ticket number (a progressive unique identifier)
Date: date of issue
Arrival: time of issue (in 24-hour format)
PaidOn: date of payment
PaidAt: time of payment (in 24-hour format).

(a) Describe how a linked list is a suitable data structure for the given scenario. [2]

Award ***[1]*** *for a feature of a linked list.*
Award ***[1]*** *for relating it to the given scenario.*

Size of a linked list is not fixed/predetermined (efficient use of memory);
Suitable because the number of objects/cars in the car park may vary greatly;

Efficient addition of elements to the linked list;
Suitable because cars arrive at the car park in any order;

When a car arrives and the car park is not full, a ticket is issued, the entrance barrier is raised and the display is updated.

Payment of a ticket at a machine updates both the ticket and its object representation held in the linked list. The car must exit the car park within 10 minutes.

At the exit barrier the ticket is checked, and this makes the barrier rise and updates the display.

(b) For the given scenario, identify:

(i) **one** example of two processes that could occur concurrently [1]

Award ***[1 max]***.
Raising both barriers;
Payment of the ticket at the machine, with any other process;

(ii) **one** example of two processes that could not occur concurrently. [1]

Award ***[1 max]***.
Incrementing and decrementing the display cannot happen concurrently;
Updating the linked list/collection following a ticket issue and a payment;
The barrier raising and the ticket being checked;
etc.

(c) State the condition that needs to be checked on the ticket when a car leaves the car park. [1]

Award ***[1 max]***.
The ticket has been paid at most 10 minutes ago;
Accept formulations such as `CurrentTime - PaidAt <= 10`;

A car arrives at the entrance barrier while another car is at the exit barrier.

(d) Explain the order in which the operations for raising the barriers and updating the display should be performed, to ensure a correct and efficient management of the car park. [3]

Award up to ***[3 max]****.*
Award ***[1]*** *for evidence that operations on exit & display are* ***before*** *the operations at entrance & display.*
Award ***[1]*** *for* ***correct sequence*** *of operations at the exit.*
Award ***[1]*** *for* ***correct sequence*** *of operations at the entrance.*

At the exit barrier
(Check ticket)→raise exit barrier→increment display;

Then at the entry barrier
(Issue ticket) →decrement display→ raise entry barrier;

Upon payment, the *PaidOn* and *PaidAt* fields are populated in the corresponding object, without removing it from the linked list.

(e) Outline **one** implication of this choice of design in terms of efficiency. [2]

Award up to ***[2 max]****.*
Example answer (space)
Eventually the collection is too large and it could run out of memory;
This impacts on the issuing of further tickets;

Example answer (time)
Sequential access and the growing linked list means that accessing the node takes longer;

This impact on the time needed for issuing/checking tickets (cars are waiting longer in the queue to leave/enter the car park);

The car park rules enforce a short-stay policy. Staying in the car park for up to 2 hours is allowed, and is subject to two possible **fees**. Staying in the car park for durations longer than two hours is subject to three possible **fines**, in addition to the original fee, up to a maximum price for each day. Tickets are paid in Euros (EUR).

The possible fees and fines are stored in a two-dimensional (2D) array, *RULES*.

		Up to 30 minutes ↓	*Up to 2 hours* ↓	*Maximum daily price* ↓
RULES		[0]	[1]	[2]
Fees →	[0]	0.50 EUR	3.00 EUR	-------
Fines →	[1]	5.00 EUR	15.00 EUR	30.00 EUR

For example:

- staying in the car park for 40 minutes costs 3.00 EUR
- staying in the car park for 3 hours costs 3.00 + 15.00 = 18.00 EUR
- any stay in the car park that exceeds 4 hours costs 30.00 EUR
- a stay that spans two consecutive days, regardless of duration, costs 60.00 EUR.

(f) Construct the steps of an algorithm that calculates the amount that a ticket is to be charged. [5]

Award marks as follows up to ***[5 max]****.*
Award ***[1]*** *for calculating excess dates (XD) (as a step only – no precise code required).*
Award ***[1]*** *for a correct calculation of payment due for fines lasting more than one day, upon test.*
Award ***[1]*** *calculating parking duration (DRN) within the same day (as a step only, no precise code required).*
Award ***[1]*** *for correct calculation of payment due for parking duration lasting less than 2 hours (within the same day) (CASE 1 and CASE 2).*
Award ***[1]*** *for correct calculation of payment due for parking duration lasting more than 2 and less than 4 hours (within the same day) (CASE 3 and CASE 4).*
Award ***[1]*** *for correct calculation of payment due for parking duration lasting more than 4 hours (within the same day).*

For example:
Compute excess dates (XD), using arrival date (AD) and current date (CD)
(*eg* XD = CD–AD);

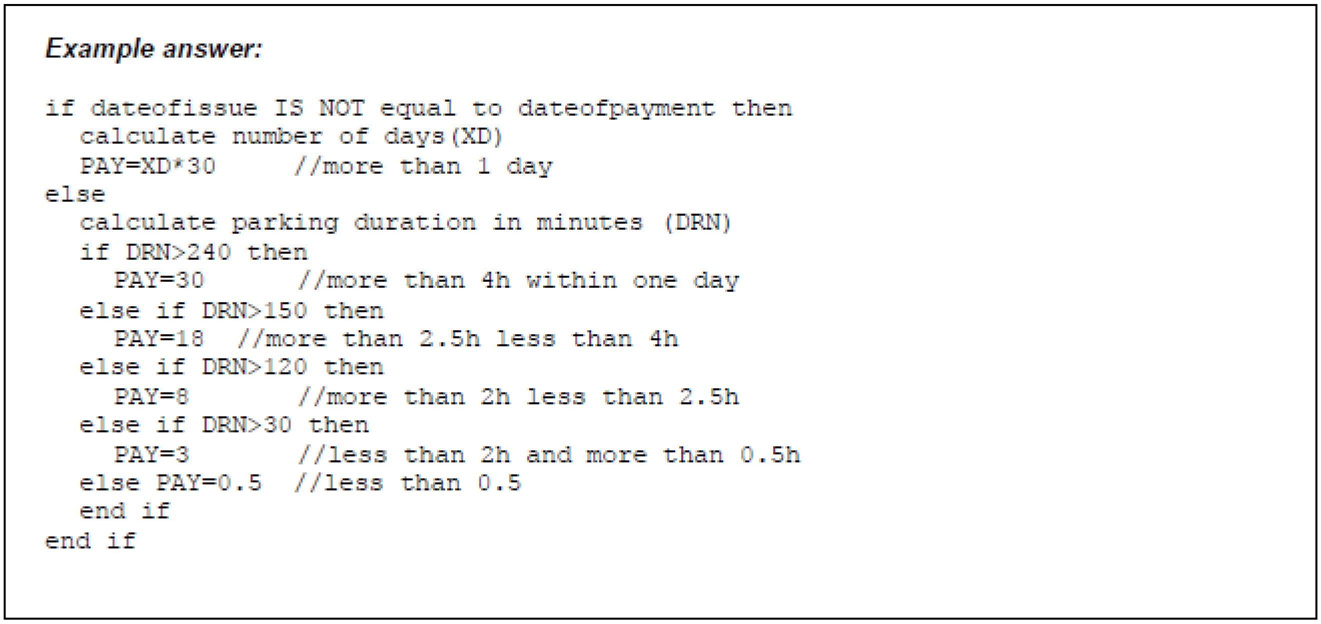
If XD ≠ 0 then PAY= XD*RULES[1][2] (accept PAY = XD*30)
(*ie* if the dates are not the same, the amount to pay is the maximum fine for XD days);

Else compute parking duration (DRN), using arrival time (AT) and current time (CT)
(*eg* DRN = CT–AT);

After suitable transformation into minutes;

```
Case (1): DRN =< 30 then PAY = RULES [0][0]
          (accept 0.50 EUR)
Case (2): 30<DRN=<120 then PAY = RULES [0][1]
          (accept 3.00 EUR)
Case (3): 120<DRN=<150 then PAY = RULES [0][1] + RULES [1][0]
          (accept 3+5 = 8 EUR)
Case (4): 150<DRN=<240 then PAY= RULES [0][1] + RULES [1][1]
          (accept 3+15= 18 EUR)
Case (5): 240<DRN then PAY = RULES [1][2]
          (accept 30 EUR)
```

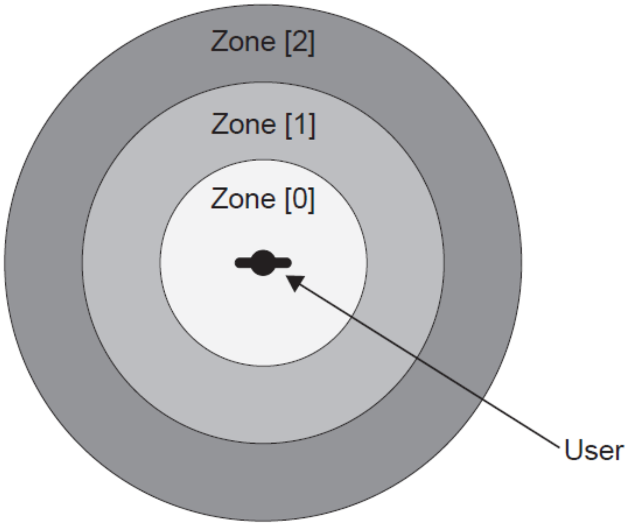
Notes: *Other solutions are possible, award marks accordingly.*
It is not necessary to refer to RULES (and relative notation) as a double array.
Accept constructions with flowcharts or pseudocode.
Accept calculations that make use of div and mod, provided they are correct and respect the specification of a daily-based fee system.
Accept DRN presented in hours.
Award marks for logic, not for the syntax



Example answer:

```
if dateofissue IS NOT equal to dateofpayment then
  calculate number of days(XD)
  PAY=XD*30     //more than 1 day
else
  calculate parking duration in minutes (DRN)
  if DRN>240 then
    PAY=30      //more than 4h within one day
  else if DRN>150 then
    PAY=18  //more than 2.5h less than 4h
  else if DRN>120 then
    PAY=8       //more than 2h less than 2.5h
  else if DRN>30 then
    PAY=3       //less than 2h and more than 0.5h
  else PAY=0.5  //less than 0.5
  end if
end if
```

14. *LookUpLunch* is an app for a Smartphone that can be used to search for restaurants located in zones of increasing distance from the user's current position. The diagram shows the user and zones as they would appear on a map of the area.



A search in Norway produced the following table, *RESULTS*, which shows the number of restaurants in each zone. *RESULTS* also displays the average price for a meal, expressed in the local currency (Norwegian Krone, NOK).

RESULTS

	NOK 90	NOK 200	NOK 400
Zone [0]	3	10	7
Zone [1]	4	6	3
Zone [2]	2	3	1

(a) Using the table, state the total number of restaurants in the zone furthest from the user. [1]

6

(b) Suggest how the zones are calculated and displayed on the map using GPS based technology. [4]

Award *up to* ***[2 max]*** *for computing the user's location.*
Award up to ***[2 max]*** *for drawing the zones on the map.*

The GPS receiver in the Smartphone picks up the signals from 3 satellites (at least 3);
The signals transmitted are: time of transmission, coordinates of the satellite;
The receiver knows when the signal was received;
Calculate positioning though equation resolution on a sphere;

Zones are displayed relative to the user's current position;
And the current scale of the map;
For each distance required, a circle is drawn (on the map) to define the zone, centred on the current GPS position;

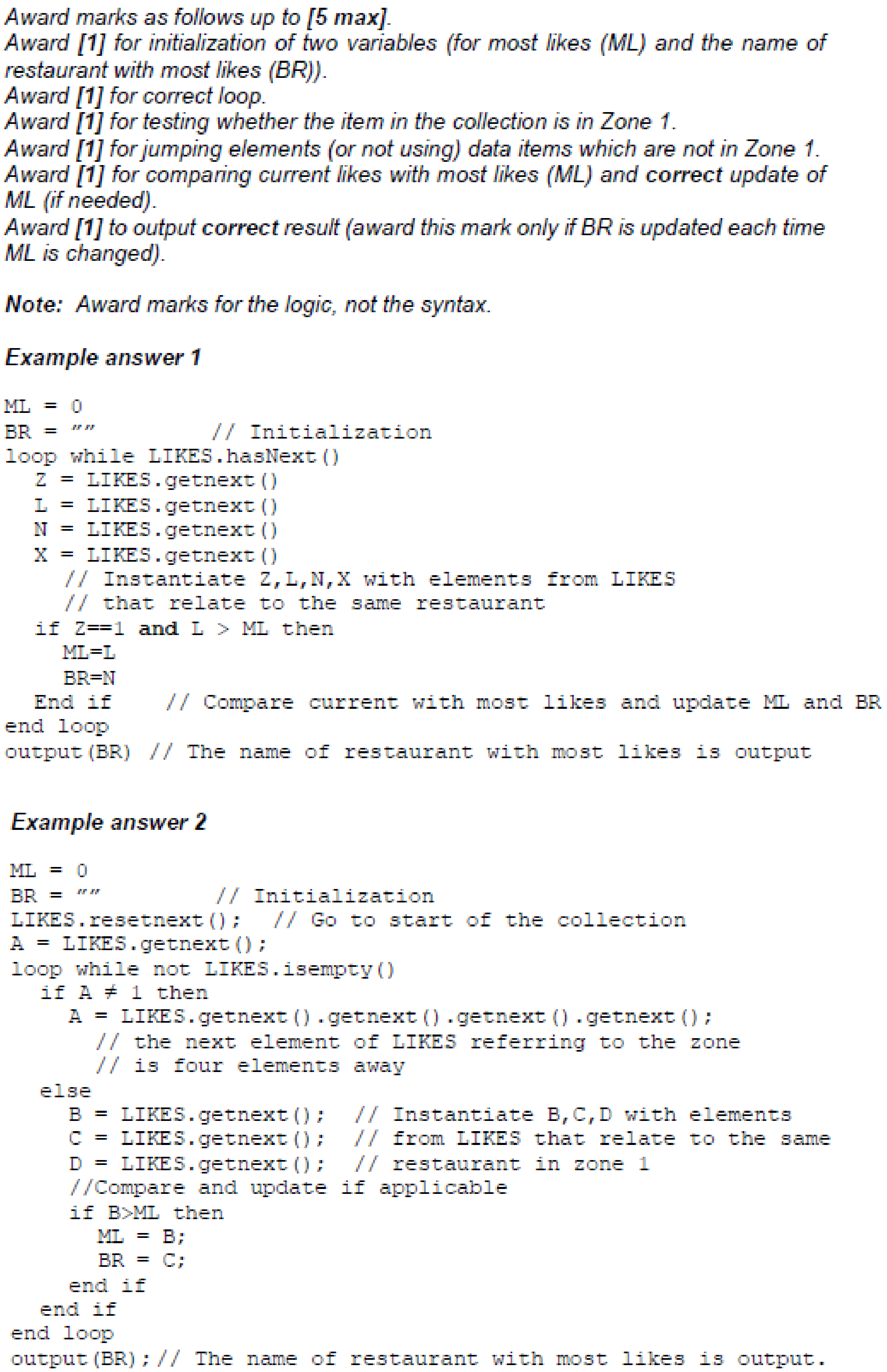
Another Smartphone app that is linked to *LookUpLunch* collects customers' reviews for restaurants.

A review consists of whether a customer likes the restaurant, and a rating of cheap (*C*), medium (*M*) or expensive (*E*). The app combines all of the reviews to produce a single letter rating (*C*, *M* or *E*) and a total number of likes for the restaurant.
As part of the internal representation of the app, the collection *LIKES* is used. Some of the data items contained in *LIKES* are shown below. Each individual data item is separated by a comma.

0,26,TomHus,M,1,14,GladLaks,E,2,1,MerPoteter,C,1,15,Linie,E,0,2,Mezze,M…

The restaurant GladLaks, underlined as an example, is located within zone [1]. Based on the reviews, this restaurant has 14 likes and is expensive.

(c) Construct an algorithm that outputs the name of the restaurant with the most likes in zone [1]. [5]



*Award marks as follows up to **[5 max]**.*
*Award **[1]** for initialization of two variables (for most likes (ML) and the name of restaurant with most likes (BR)).*
*Award **[1]** for correct loop.*
*Award **[1]** for testing whether the item in the collection is in Zone 1.*
*Award **[1]** for jumping elements (or not using) data items which are not in Zone 1.*
*Award **[1]** for comparing current likes with most likes (ML) and **correct** update of ML (if needed).*
*Award **[1]** to output **correct** result (award this mark only if BR is updated each time ML is changed).*

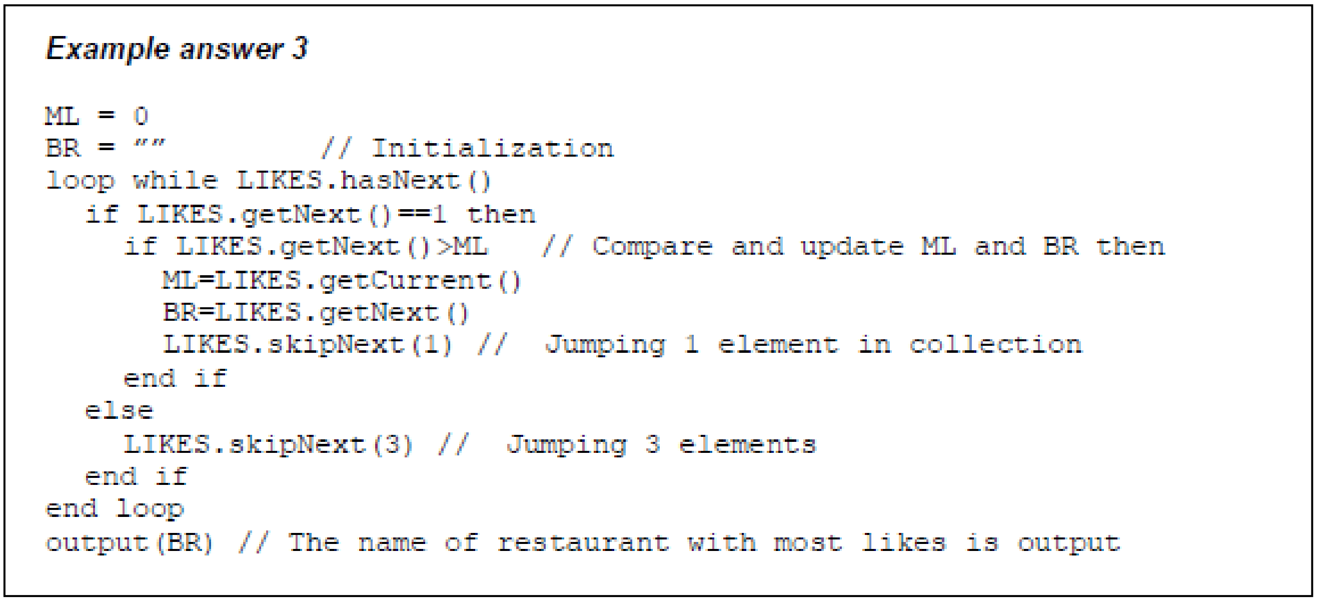
***Note:** Award marks for the logic, not the syntax.*

Example answer 1

```
ML = 0
BR = ""          // Initialization
loop while LIKES.hasNext()
   Z = LIKES.getnext()
   L = LIKES.getnext()
   N = LIKES.getnext()
   X = LIKES.getnext()
      // Instantiate Z,L,N,X with elements from LIKES
      // that relate to the same restaurant
   if Z==1 and L > ML then
      ML=L
      BR=N
   End if    // Compare current with most likes and update ML and BR
end loop
output(BR) // The name of restaurant with most likes is output
```

Example answer 2

```
ML = 0
BR = ""          // Initialization
LIKES.resetnext();  // Go to start of the collection
A = LIKES.getnext();
loop while not LIKES.isempty()
   if A ≠ 1 then
      A = LIKES.getnext().getnext().getnext().getnext();
         // the next element of LIKES referring to the zone
         // is four elements away
   else
      B = LIKES.getnext();  // Instantiate B,C,D with elements
      C = LIKES.getnext();  // from LIKES that relate to the same
      D = LIKES.getnext();  // restaurant in zone 1
      //Compare and update if applicable
      if B>ML then
         ML = B;
         BR = C;
      end if
   end if
end loop
output(BR); // The name of restaurant with most likes is output.
```



Example answer 3

```
ML = 0
BR = “”          // Initialization
loop while LIKES.hasNext()
  if LIKES.getNext()==1 then
    if LIKES.getNext()>ML    // Compare and update ML and BR then
      ML=LIKES.getCurrent()
      BR=LIKES.getNext()
      LIKES.skipNext(1) //  Jumping 1 element in collection
    end if
  else
    LIKES.skipNext(3) //  Jumping 3 elements
  end if
end loop
output(BR) // The name of restaurant with most likes is output
```

(d) By making use of binary trees and the collection *LIKES*, explain how a list could be produced that shows the restaurants in order of zone and then, within each zone, in order of popularity. [3]

Award marks as follows up to ***[3 max]****.*
Award ***[1]*** *for associating trees to zones.*
Award ***[1]*** *for explaining how to use the collection's elements to build the trees.*
Award ***[1]*** *for inserting nodes under some criterion.*
Award ***[1]*** *for visiting the trees in such a way as to produce the ordered list.*

For example:
Create three binary trees (one for each zone), by taking 4 elements at a time from `LIKES`;
The element “zone” identifies the tree **and** the other three elements make a new node in that tree;
Nodes are added depending on “likes” so that left child =< (accept “<”) the (subtree) root, less than right child;
Do an inorder traversal on the three trees to get three lists, then join them together;

Individual restaurants are able to use this app to see reviews from their customers.

(e) Describe one disadvantage of the use of feedback from social networking in relation to business. [2]

*Award **[1]** for **one** disadvantage and **[1]** for an elaboration, up to **[2 max]**.*

For example:
Feedback/reviews might not correspond to trustworthy judgement;
And call something cheap that is not cheap / something good that is not so good;
The same person might use more than one identity to add weight to opinions;
There is a market for fake reviews, and people may even be paid to author them;
It is hard to counterbalance negative and not trustworthy feedback for the business because of the complexity of the review system;

***Note:** Accept other plausible answers.*

15. The letters $F_0, F_1, F_2, \ldots, F_N, \ldots,$ where N ≥ 0, are used to identify the *N* th term of the sequence of Fibonacci numbers that starts as follows.

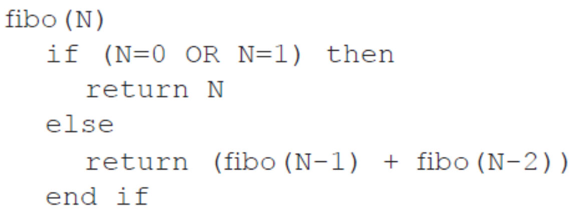
```
0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , …
```

With the exception of the leading 0 and 1 (the zeroth term and 1st term), the terms in the sequence are the sum of the two preceding terms. For example, F_5 is the 5th term of the sequence, which is 5, and is the sum of the 3rd and 4th terms, which are 2 and 3 respectively.

(a) State the value of the 8th term in the sequence. [1]

21

The following method, *fibo(N)*, generates the *N* th term in the sequence. The *return* statement returns the value that the method generates.



```
fibo(N)
  if (N=0 OR N=1) then
    return N
  else
    return (fibo(N-1) + fibo(N-2))
  end if
```

(b) Trace *fibo(4)*, showing the different levels of recursion. [3]

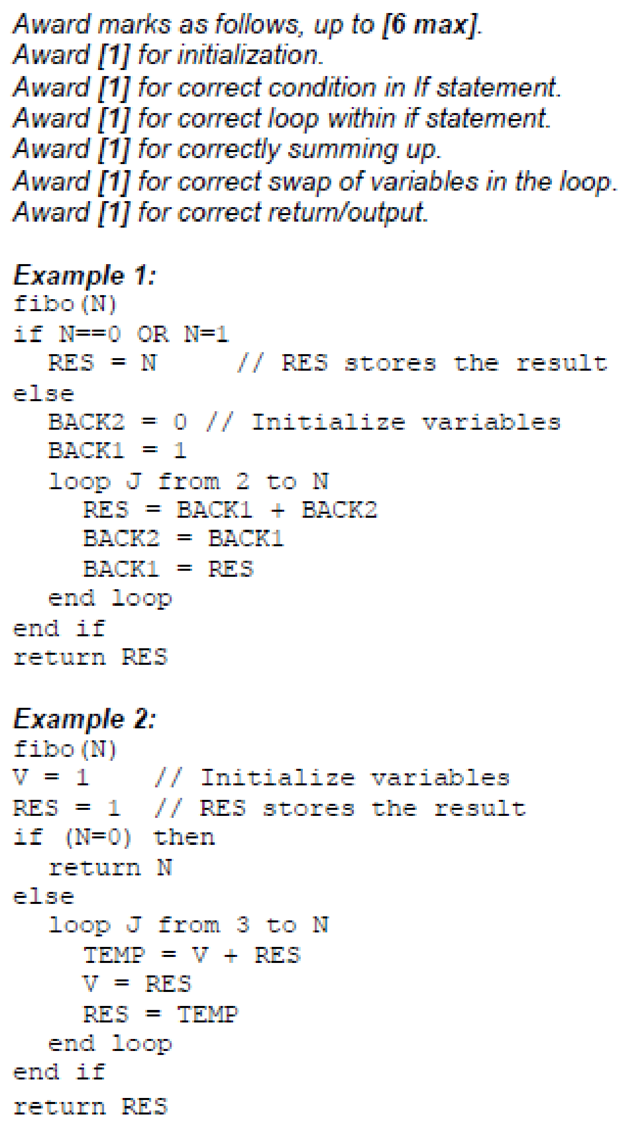
Award marks as follows up to ***[3 max]****.*
Award ***[1]*** *for evidence of* ***two recursive calls*** *of* `fibo(2)` *or of two recursive calls of* `fibo(0)`*.*
Award ***[1]*** *for evidence of* ***three recursive calls*** *of* `fibo(1)`*.*
Award ***[1]*** *for correctly returning values* `0` *and* `1` *for* `fibo(0)` *and* `fibo(1)` *when needed.*

For example:

```
fibo(4)
= (fibo(3)+fibo(2))
= (fibo(2)+fibo(1)) + (fibo(1)+fibo(0))
= ((fibo(1)+fibo(0)) +1) + (1+0)
= ((1+0)+1) + (1+0)
= 3
```

Note: *No marks shall be given if the answer 3 is provided without any tracing, nor if the intermediate values instead of calls to* `fibo()` *are given; this is a tracing question.*
The order of evaluation of intermediate steps may be different from the one presented in the solution here.

(c) Construct a non-recursive algorithm to generate Fibonacci numbers. [6]



Award marks as follows, up to ***[6 max]****.*
Award ***[1]*** *for initialization.*
Award ***[1]*** *for correct condition in If statement.*
Award ***[1]*** *for correct loop within if statement.*
Award ***[1]*** *for correctly summing up.*
Award ***[1]*** *for correct swap of variables in the loop.*
Award ***[1]*** *for correct return/output.*

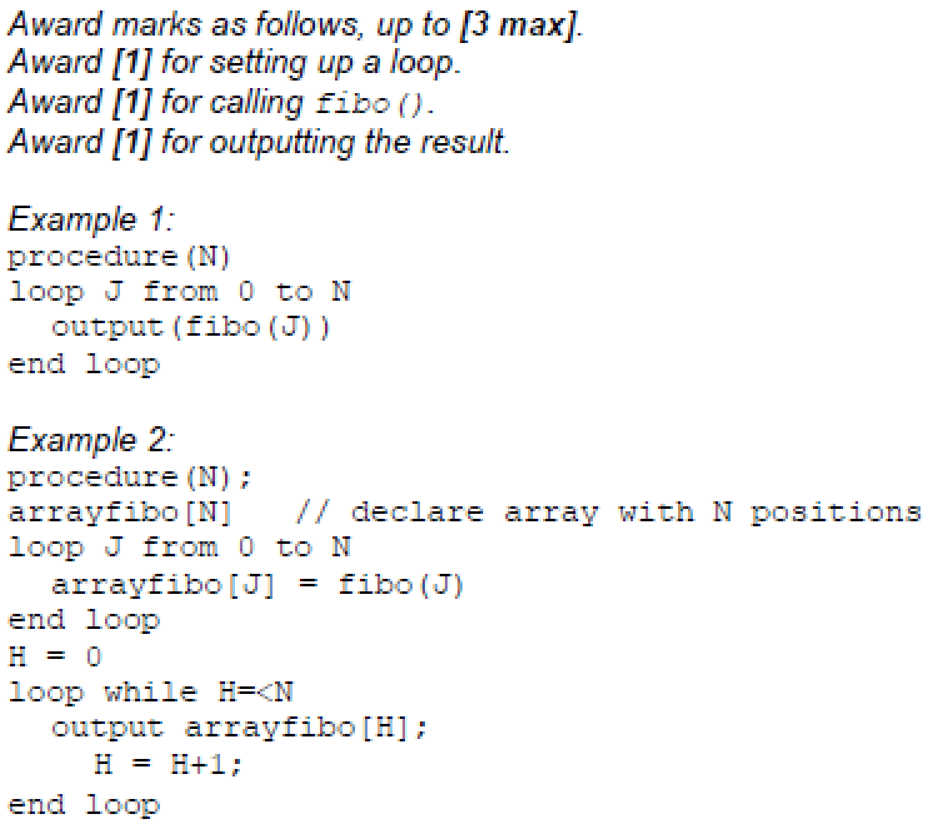
Example 1:

```
fibo(N)
if N==0 OR N=1
   RES = N     // RES stores the result
else
   BACK2 = 0 // Initialize variables
   BACK1 = 1
   loop J from 2 to N
      RES = BACK1 + BACK2
      BACK2 = BACK1
      BACK1 = RES
   end loop
end if
return RES
```

Example 2:

```
fibo(N)
V = 1     // Initialize variables
RES = 1   // RES stores the result
if (N=0) then
   return N
else
   loop J from 3 to N
      TEMP = V + RES
      V = RES
      RES = TEMP
   end loop
end if
return RES
```

(d) Construct an algorithm that will **output** the first *N* terms of the sequence. You should use *fibo()*, the method defined above. [3]



Award marks as follows, up to ***[3 max]****.*
Award ***[1]*** *for setting up a loop.*
Award ***[1]*** *for calling* `fibo()`*.*
Award ***[1]*** *for outputting the result.*

Example 1:

```
procedure(N)
loop J from 0 to N
   output(fibo(J))
end loop
```

Example 2:

```
procedure(N);
arrayfibo[N]    // declare array with N positions
loop J from 0 to N
   arrayfibo[J] = fibo(J)
end loop
H = 0
loop while H=<N
   output arrayfibo[H];
      H = H+1;
end loop
```

Note: *Accept comparable solutions, also in flowcharts, provided they are correct.*

Recursive programs written in high level languages require the use of particular structures to support their execution.

(e) Describe how a stack is usually employed in the running of a recursive algorithm. [2]

The current environment (*eg* values/local variables/current address/registers) PUSHED onto the stack when a new recursive call is met;

To be POPPED OFF the stack when the recursive subprogram is completed.