May 2014 HL P1

Section A

1. Identify **two** features that need to be considered when planning a new computing system for an organization. [2]

*Award up to **[2 marks max]**.*
Roles/activities of the users (*eg* permissions, security, partitions, collaborative work);
Resources (HW and SW equipment) appropriate for the organization;
Costs/budget limits;
Delivery time;
Compatibility with the old system (data);

(Other acceptable answers are possible, from the economic, operational and technical perspective.)

2. Explain what is meant by *beta testing*. [2]

*Award up to **[2 marks max]**.*
Testing prior to product's full release / last stage of testing;
To see if it works properly / complete functionality / usability;
Performed by end users (not by designers);

3. Describe **one** advantage and **one** disadvantage of using observations to gather information when planning a new system. [4]

*Award up to **[4 marks max]**.*
*Award **[1 mark]** for the identification of an advantage and **[1 mark]** for explaining the advantage.*
*Award **[1 mark]** for the identification of a disadvantage and **[1 mark]** for explaining the disadvantage.*

Example Advantages:
Can highlight aspects that are not detected in questionnaires/interviews;
So the observer can help produce more detailed reports;

Observations may be more reliable than interviews;
Because they can reveal what people actually do instead of what they say they do;

Example disadvantages:
Time consuming / expense;
The observer might need to observe a complete cycle *etc* which could last a significant amount of time;

If the observations are made by only one person, they may be biased;

Observations may be unreliable;
Because people act differently when they know they are being watched.
(Accept formulations that express similar or plausible ideas.)

4. Outline **one** usability issue associated with the design of mobile devices. [2]

*Award **[1 mark]** for identifying an issue and **[1 mark]** for an explanation.*
Size of screen;
Therefore difficult to see / use (in poor light);

Size of keys;
Therefore difficult to access functions;

Battery life;
May need to recharge regularly;

Touch screen keys on tablets *etc*;
Lack of tactile feedback;

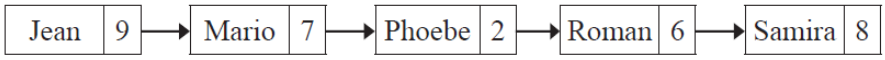
5. Distinguish between the use of **two** types of primary memory. [2]

*Award up to **[2 marks max]**.*
*Award **[1 mark]** for identifying **two** types of primary memory.*
*Award **[1 mark]** for the use of **each** type of the memory identified ×2.*

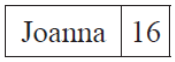
RAM stores data and instructions currently in use
ROM stores permanent instructions
Cache stores frequently used instructions

*(Award **[1 mark]** if only general scheme of CPU is given.)*

6. Consider the following linked list which is maintained in alphabetical order.



With the aid of diagrams, explain how the node



would be inserted into the linked list. [3]

Initially compare with node pointed to by the head;
(If not correct place) move through list using pointers until correct alphabetical position is found;
Adjust pointers accordingly;

(Drawings are acceptable, but award marks only if they clearly show how pointers are correctly rearranged, following the three guidelines above.)

7. Outline how a colour can be represented in a computer. [2]

A colour will be split into three components *(Accept RGB as an example)*;
Each component will be assigned a certain number of bytes;

8. Identify **two** key features of a peer-to-peer (P2P) network. [2]

*Award up to **[2 marks max]**.*
Examples of features:
No central server;
Resources are more widely available (storage, bandwidth, computing power);
Redundancy/recovery;
Supports file sharing for collaborative work;

9. Outline the role of paging in the management of primary memory. [2]

*Award up to **[2 marks max]**.*
It is used in the formation of virtual memory / use of secondary memory;
To increase the amount of primary memory;
Memory divided into (tagged) "pages";
Which are then transferred in and out as required;

10. Outline **two** distinct features of autonomous agents. [4]

*Award **[1 mark]** for a feature and **[1 mark]** for a description, for two features, up to **[4 marks max]**.*

Feature: Autonomy;
Description:
Agents *activate* alone for a task and are not invoked for a task;
Agents can *select the task* themselves (based on priorities or goal-directed search) without human intervention;

Feature: Reactive behavior;
Description:
Agent senses the environment in which it is, and decides what to do reacting on its perceptions;

Feature: Concurrency/sociality;
Description:
Agents can interact with other agents through communication, in different modes: coordination, cooperation, competition;

Feature: Persistence;
Description:
The code describing an agent runs continuously like a process, and is not executed on demand;

Section B

11. A builder is renovating a series of apartments and is considering integrating a few electrical devices in each apartment into an automatic programmable system. One example is the integration of lighting, heating, ventilation and air conditioning.

    (a) Identify **two** groups of users that might find this integrated technology particularly appealing. [2]

*Award up to **[2 marks max]**.*
Elderly;
Disabled;
Commuters;
Accept other reasonable answers.

    (b) Discuss **two** advantages, offered by this technology, that could be used in an advertisement for the apartments. [4]

*Award **[1 mark]** for advantage and **[1 mark]** for explanation, for two items, up to **[4 marks max]**.*
Improved convenience;
One can better control times of functioning, hence costs;

Improved comfort;
One can program the functions according to their specific needs;

Energy efficiency;
One can program/plan the functions based on the surrounding environment and reduce energy waste (interconnected systems);

Safety;
Programmed in a way to avoid electric overload and faults;

Accept other reasonable answers

(c) Evaluate **two** ways users can access the functionality of the integrated system at home. [6]

*Award up to **[6 marks max]**.*
*Award up to **[3 marks max]** for each of the two ways.*
*Award **[1 mark]** for the way of access and up to **[2 marks]** for two additional points, which could be advantages or disadvantages.*
Examples include fixed/non-fixed installations, digital/analogue.

Computer based/TV-based control;
The user always knows where the control is;
Because the device may not be portable;
Inconvenient if computer/TV already being used by someone else;

Touch screen/keypads in fixed installation;
Ergonomic gadget/small dimensions/cheap device;
Wide range of different designs to fit different locations in a house;
Difficult to lose;
Inconvenient to operate some devices by having to go to a fixed panel/not suitable for extended use to operate TVs etc;

Smartphone;
Portable/useful for some groups of users (limited mobility);
Can be easily extended to include other devices;
Could be lost or misplaced;
Requires internet/Wi-Fi/signal to operate;

Accept other suitable answers.

The same technology is adapted and used for intensive chicken farming; in this context a decentralized control is preferred.

(d) Describe how this could be achieved. [3]

*Award up to **[3 marks max]**.*
Transmission: integrated wiring or internet or wireless;
Requires: extended/dedicated network and hw/sw for protocols/transmissions and sensors/actuators;
Use: The farmer can vary the parameters/environmental conditions from home at any time and better concentrate on other activities (*eg* trade with KFC);

12. An international organization has offices located across several countries. For some of its activities, for example human resource management, it has been decided to adopt a "Software-as-a-Service" (SaaS) solution in order to keep the running costs low.

(a) Describe the features of SaaS. [3]

SW necessary for the activities is in the cloud;
Access to SW is with thin client (terminal/computers) by web browsing (on the extranet);

(b) Discuss the limitations of SaaS in relation to security. [6]

*Award up to **[6 marks max]**.*
Security in storage;
Data is stored in the server of the service provider;
The organization has no direct control of its data;
Legislation in the country of the provider may be weaker than in the user's country;
Cases of provider's corruption/bankruptcy/data loss are a risk to the organization;

Security in transmission;
Applications running in-site may require data in SaaS;
Hence longer transmission times and higher risk of failure/attack/interception;

Each office makes some data available to external customers through the use of an extranet and allows employees to work from home through a VPN.

(c) Define the term extranet. [2]

*Award up to **[2 marks max]**.*
An external extension to a company's local network;
Limited access;
Uses internet protocols;

(d) Distinguish between a VPN and an extranet. [4]

VPN authenticates the sender before (establishing the tunnel);
VPN access is always encrypted, whereas extranet has limited encryption;
VPN transmission is always encrypted;
VPN users have access to everything whereas extranet users only have access to (enabled) specific services;

13. The faceplate of a car stereo has six buttons for selecting one of six preferred radio stations. As part of the internal representation of a microprocessor there is an array with six positions, carrying the information about the radio frequencies, as follows.

Radio

[0]	[1]	[2]	[3]	[4]	[5]
100.4	88.7	90.2	104.5	93.8	106.2

(a) State the information at *Radio[2]*. [1]

90.2

(b) Outline how a numerical frequency could be stored in a fixed-length string. [2]

Frequencies less than 100 take a 0 on the left (*eg* 88.7 becomes 088.7);
Convert each digit into a char to get a string;
Allow the "dot" to be omitted in the interpretation. There is always only one

(c) Construct an algorithm in pseudocode that calculates the range of frequencies (*ie* the difference between the highest and lowest frequencies) of any set of six selected radio stations. [6]

*Award up to **[6 marks max]**.*

***Example answer** (searches for the min and max, and then the range is calculated)*
*Award **[1 mark]** for each of the following*
Initialization;
Loop;
Correct if statement (min);
Correct if statement (max);
Compute the range;
Output the range;

```
MIN = Radio[0]
MAX = Radio[0]
K=1
loop while K<=5
      if Radio[K]<MIN then
            MIN=Radio[K]
      else if Radio[K]>MAX then
            MAX=Radio[K]
      endif
      K=K+1
endloop
RANGE=MAX-MIN
output RANGE
```

***Example answer** (sorts the array Radio, and then the range is calculated, any sorting algorithm is acceptable)*
*Award **[1 mark]** for each of the following*
Idea of nested loops;
Correct loops;
Correct comparison;
Correct exchange;
Compute the range;
Output the range;

```
loop for  K=0 to 4
     loop for  J=0 to 4
      if Radio[J]> Radio[J+1]
       then
          swap Radio[J]and Radio[J+1]
      endif
     endloop
endloop
RANGE= Radio[5]- Radio[0]
output RANGE
```

The two-dimensional array *Stats* provides an indication of how often a specific station is listened to by the user. For each button in the faceplate it records how often it has been clicked in the last 48 hours. *Stats* is ordered by the second column.

Stats

5	13
4	9
0	8
3	4
1	3
2	2

Both *Radio* and *Stats* are used by a procedure that allows the user to access the radio frequencies that are listened to most often, as recorded in *Stats*, by flicking a lever on the steering wheel. The frequencies are accessed cyclically, *ie* after the least used frequency the procedure returns to the most used. For this reason a queue **Q** is used.

(d) Construct an algorithm in pseudocode that, by using the structures *Radio* and *Stats*, performs the following steps:

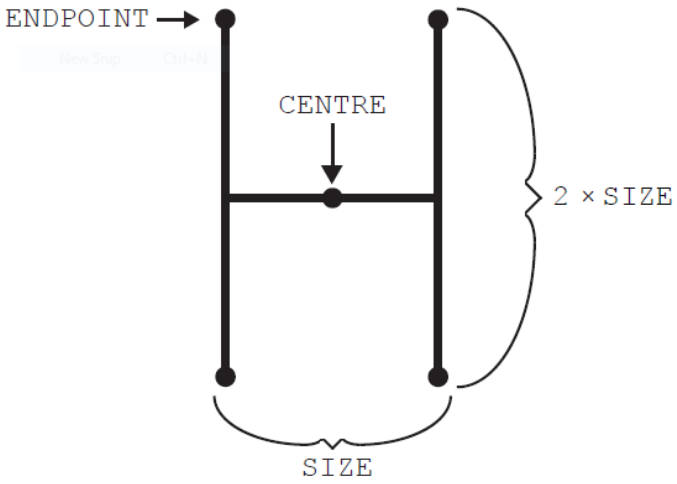
- it inserts the radio frequencies in the queue **Q**, following the actual order of preference; and then
- it uses the queue **Q**, cyclically, to output an element each time the lever is flicked. [6]

*Award up to **[6 marks max]**.*
Possible solution below, marks to be awarded following the comments in code (there are 7 marks at least).

```
// Q is given and initially empty
I=0
loop while I<6                  //across all rows
     ITEM=Stats[I,0]            //retrieve the button
     Q.enqueue(Radio(ITEM))     //enqueue the frequency
     I=I+1
endloop

if input(Flick) then            //if a flicking occurs
     TEMP=Q.dequeue()           //dequeue an element from Q
     output(TEMP)               //to output it and
     Q.enqueue(TEMP)            //circular use of the queue
endif
```

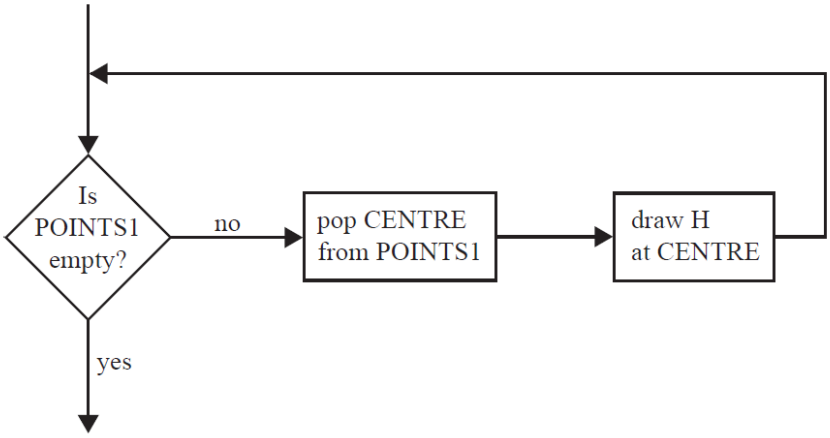
14. Consider the following diagram and pseudocode for drawing on a display screen.




```
ENDPOINTS = drawH(CENTRE, SIZE)
```

The method *drawH(CENTRE, SIZE)* will draw an "H" located at *CENTRE* with width of *SIZE* and height of *2 × SIZE*, as shown. It returns an array containing the four **endpoints** of the vertical lines.

In the following flowchart, *POINTS1* is a stack.



(a) Construct pseudocode corresponding to the flowchart. [3]

*Award up to **[3 marks max]**.*
*Award **[1 mark]** for a loop (while).*
*Award **[1 mark]** for correct condition/logical expression.*
*Award **[1 mark]** for correct operations.*

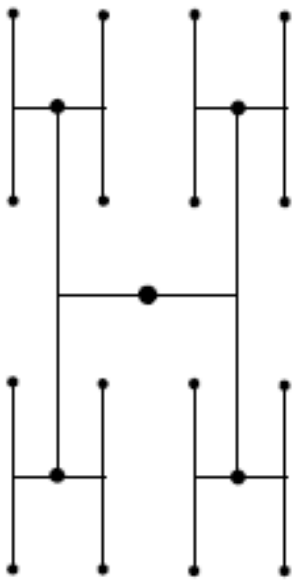
```
loop while NOT POINTS1.isEmpty()
    CENTRE = POINTS1.pop()
    drawH(CENTRE, SIZE)
end loop
```

(b) Construct the drawing that would be produced by the flowchart on page 6 if it is preceded by the following steps.

```
SIZE = 20
CENTRE = the middle of the user's display
POINTS1 is a stack, initially empty
ENDPOINTS = drawH(CENTRE, SIZE)
loop COUNT from 0 to 3
  POINTS1.push(ENDPOINTS[COUNT])
end loop
SIZE = SIZE / 2
```

[2]

*Award **[1 mark]** for correct number and placement of H's;*
*Award **[1 mark]** for reasonable attempt to show reduced size of successive H's;*



The pattern of drawing a new set of H's, which have a *SIZE* value that is half the *SIZE* value of the previous H, can be repeated. Each set of H's of the same size is called a generation.

(c) Construct an algorithm that will draw an initial H in the centre of the display and three generations after that. [6]

Award up to ***[6 marks max]*** *as follows:*
[1 mark] *for the idea that there needs to be a loop that executes once for each generation,* ***with an additional [1 mark]*** *if it is implemented correctly, for* ***[2 marks max]****.*

[1 mark] *for the idea that the endpoints returned while drawing one generation must be stored for use in drawing the subsequent generation,* ***with an additional [1 mark]*** *if correctly implemented, for* ***[2 marks max]****.* ***Note:*** *Any valid storage mechanism is permissible ie it does not have to be a stack.*

[1 mark] *for correctly drawing a generation of H's.*

[1 mark] *for correctly scaling the size for each generation.*

Example answer:

```
////////////////////
SIZE = 20                                  //do not award marks
CENTRE = the middle of the user's display
POINTS1 is a stack, initially empty        //for this part of the
ENDPOINTS = drawH( CENTRE, SIZE )          // of the algorithm
loop COUNT from 0 to 3
     POINTS1.push( ENDPOINTS[COUNT] )      // it is given
end loop
SIZE = SIZE / 2                            // in the question paper
////////////////////

POINTS2 is a stack, initially empty
loop GENERATION from 1 to 3
     loop while NOT POINTS1.isEmpty()
          CENTRE = POINTS1.pop()
          ENDPOINTS = drawH(CENTRE, SIZE)
          loop COUNT from 0 to 3
               POINTS2.push( ENDPOINTS[COUNT] )
          end loop
     end loop
     POINTS1 = POINTS2
     empty the POINTS2 stack
     SIZE = SIZE / 2
```

(d) State how many endpoints there will be after the initial H and three generations have been drawn, without any consideration of the size. [1]

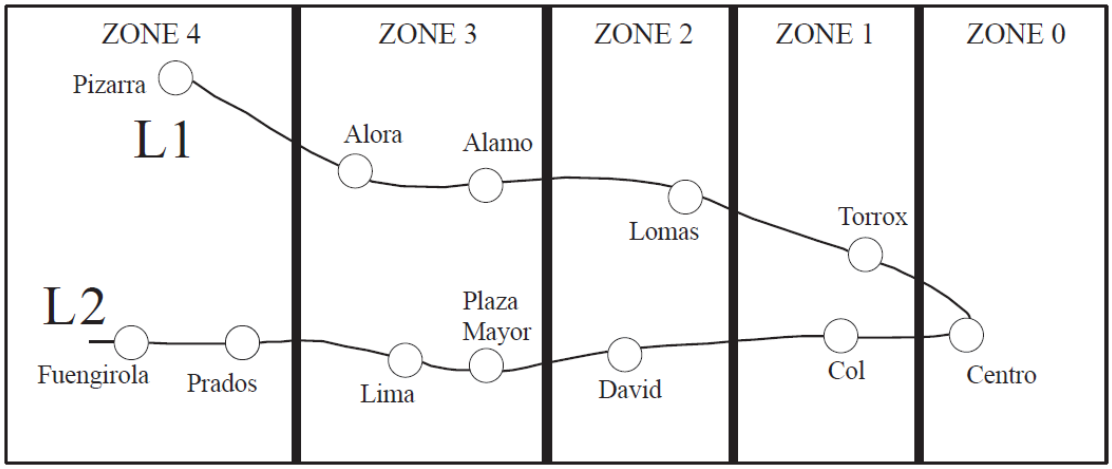
Accept any expression appearing in the following equivalences
4*4*4*4 = 4_4 = 256;

(e) Suggest how drawing this pattern of H's could be done recursively. [3]

*Award up to **[3 marks max]** for:*
identifying that the recursive algorithm proceeds downwards (showing on one parameter will suffice);
base case of recursion;
recursive call;

The recursive algorithm would use the parameters centre, size, generation counting down (not up);
When the generation reached is 0 no drawing is done;
Otherwise draws an H and calls recursively itself on the endpoints of a lower generation;

15. A suburban railway system for a large city in Southern Europe consists of two lines **L1** and **L2**, which meet at the station Centro, where passengers can change from one line to the other. The system is shown below.



Each station is located in a particular zone, and the total number of zones in which the journey takes place determines the train fare. Note, if a passenger starts in **Zone 1**, goes to **Zone 0** and then back to **Zone 1**, the journey has taken place in **three** zones. Examples of the number of zones are shown below for different journeys.

Travelling from	Travelling to	Number of zones
Lima	Plaza Mayor	1
Alora	Plaza Mayor	7
Lomas	Col	4

(a) State the number of zones in which the journey takes place when travelling from Alora to Fuengirola. [1]

8;

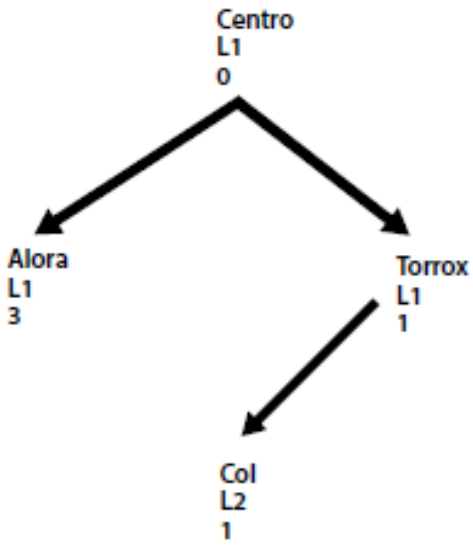
The data for each station (station name, line, zone) is stored on the system's server in the collection *TRAIN_DATA*. There are 12 stations in total. The first part of the collection is shown below.

```
Centro, L1, 0, Alora, L1, 3, Torrox, L1, 1, Col, L2, 1, ...
```

From this we can see that Alora is part of line L1 and is located in Zone 3.
At the start of each day, the data in *TRAIN_DATA* is read in to the binary tree *TREE*, in which each node will hold the data for one station. The binary tree will be used to search for a specific station's name.

(b) Sketch the binary tree after the station data from the first part of the collection, given above, has been added. [3]

Centro as root;
Station names in correct position;
All 3 items of data for each node;



The *TRAIN_DATA* collection is also used to construct the one-dimensional array *STATIONS* (which only contains the list of station names sorted into alphabetical order), where *STATIONS[0] = Alamo*.

(c) State the value of *STATIONS[4]*. [1]

David;

The two data structures (*STATIONS* and *TREE*) are now used to construct the two-dimensional array *FARES* containing the fares between stations, partly shown below. Note that the fare for travelling in each zone is €1.00.

FARES	Alamo	Alora	Centro	Col	...
Alamo	0	1.00	4.00	5.00	...
Alora	1.00	0	4.00	5.00	...
Centro	4.00	4.00	0	2.00	...
Col	5.00	5.00	2.00	0	...
...	...	...	...	...	... *etc*

(d) Calculate the fare for travelling from Torrox to Lima. [1]

5.00 (Euros);
Accept 5.

(e) Construct the algorithm that would calculate the fares for this two-dimensional array. You can make use of the following two sub-procedures:

- `TREE.getZone(STATION)` `// which returns the zone in which the station is located`
- `TREE.getLine(STATION)` `// which returns the line on which the station is located`

Your algorithm should make as few calculations as possible. [9]

*Award **[1 mark]** for each of the following 11 points, up to **[9 marks max]**.*

- *use of nested loops;*
- *use of nested loops with indices that avoid repeating calculations (as shown); (**Note:** outer loop can be to 11 if repeat calculations are avoided, with an IF statement)*
- *correct values retrieved from tree;*
- *check for same line;*
- *check if one of the stations is "Centro";*
- *check and change if negative/ use of absolute value;*
- *correct calculation for same line/one station is "Centro";*
- *correct calculation for different line;*
- *assign value to array;*
- *assign mirror value;*
- *assign value to diagonal;*

```
loop N from 0 to 10
  STATION1 = STATION[N]
  AZ = TREE.getZone[STATION1]
  AL = TREE.getLine[STATION1]
  loop M from N+1 to 11  //start index changed so as not to repeat
                         //code
    STATION2 = STATION[M]
    BZ = TREE.getZone[STATION2]
    BL = TREE.getLine[STATION2]
    if AL = BL or STATION1 = "Centro" or STATION2 = "Centro"
       then              //on same line or passing through "Centro"
          X = AZ - BZ    //number of zones where the travel takes
                         //place can be negative
          if X<0 then      //allow use of absolute
            X = -X         //or equivalent,e.g. X = abs(AZ-BZ)
          endif
          X = X+1
        else  //on different lines
        X = AZ+BZ+1
    endif
    FARES[N][M]=X   //assigns value to 2D array
    FARES[M][N]=X   //assigns mirror value
  endloop
  FARES[N][N]=0     //leading diagonal
endloop
FARES[11][11]=0     //final entry
```

[9 marks]