May 2015 HL P1

Section A

1. Describe one way that software developers can ensure that the users are aware of any available updates for their products. [2]

*Award up to **[2 marks max]**.*
*Award **[1 mark]** for communication with user – email/ pop up etc.*
*Award **[1 mark]** for method of installation of update – automatic/ link/ in list for user to install etc.*
When the software is installed and registered (a cookie is placed on the machine);
This communicates with the software developer automatically on start up;
Messages about updates are sent back to the machine and alerts are given;

OR

Send an email;
With a link to the update;

2. Construct a truth table for the following Boolean expression.

(A or B) and (not C or B) [4]

*Award **[1 mark]** for each correct pair of rows.*

A	B	C	(A or B) and (not C or B)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

3. Outline one example of the use of a virtual private network (VPN). [3]

*Award **[1 mark]** for a relevant example and **[2 marks]** for an elaboration.*

Example 1:
A business can let employees work at home / employees who travel a lot/external (non-employee) users;
Accessing the data and services (at the office);
Via secure login;

Example 2:
Using VPN, address is masked;
The location of the user is not known;
May be essential in delicate situations such as political protest groups working from their own country;
***Note:** Accept any legitimate reason for needing to be unknown.*

4. Trace the following algorithmic fragment for *N = 6*. Show all working in a trace table.

```
SUM = 0
loop COUNT from 1 to (N div 2)
     if N mod COUNT = 0 then
          SUM = SUM + COUNT
     end if
end loop
if SUM = N then
     output "perfect"
else
     output "not perfect"
end if
```

*Award **[4 marks]** as follows.*
*Award **[1 mark]** for going 3 times through the loop (with COUNT from 1 to 3).*
*Award **[1 mark]** for incrementing correctly SUM (when N mod COUNT = 0).*
*Award **[1 mark]** for the correct output ("perfect").*
*Award **[1 mark]** for showing all working in a trace table with at least three columns (eg COUNT, SUM, OUTPUT).*
*Award **the first** 3 marks for an evident trace but working not shown in a trace table.*

Example answer 1:

COUNT	N mod COUNT=0	SUM	SUM=N	output
1	TRUE	1		
2	TRUE	3		
3	TRUE	6		
			TRUE	perfect

Example answer 2:

COUNT	N mod COUNT	SUM	output
		0	
1	0	1	
2	0	3	
3	0	6	perfect

[4]

5. Use a selection sort to put the following set of numbers into order from highest to lowest.
List the results after each pass.

12 52 16 42 88 86 [3]

*Award marks as follows up to **[3 marks max]**.*
*Award **[2 marks max] for the first three passes** correct,*
*(**[1 mark]** for at least one of passes 1, 2 and 3 correct).*
*Award **[1 mark]** for correct passes 4 and 5 with no change on pass 4.*

Pass	12	52	16	42	88	86
1	**88**	**52**	**16**	**42**	**12**	**86**
2	**88**	**86**	**16**	**42**	**12**	**52**
3	**88**	**86**	**52**	**42**	**12**	**16**
4	**88**	**86**	**52**	**42**	**12**	**16**
5	**88**	**86**	**52**	**42**	**16**	**12**

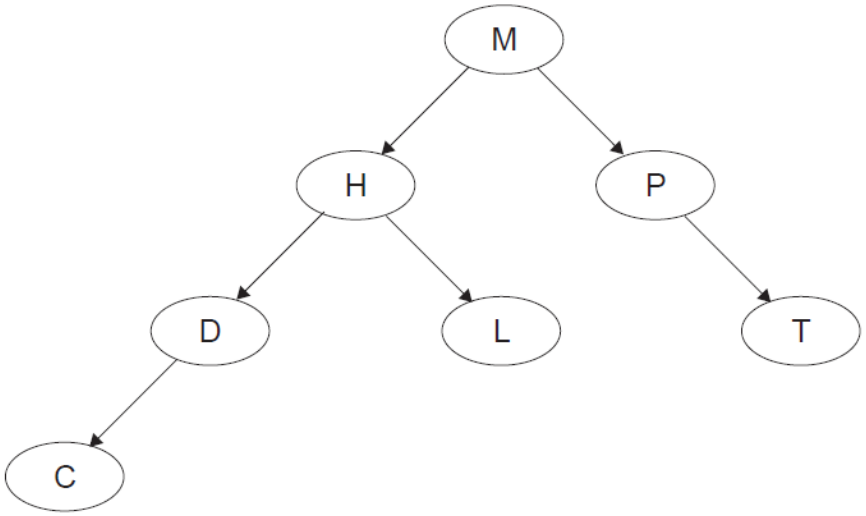
6. Describe how a GPS system can identify the position of a person. [3]

*Award up to **[3 marks max]**.*
GPS works by communication with satellites;
By knowing the position of the satellite (sent to GPS device);
And calculating the time difference between satellites;
The position of the device can be calculated;

7. Describe one way that the operating system of a networked workstation hides the complexity of the network from the user. [3]

*Award **[1 mark]** for an appropriate use for the user, **[2 marks]** for an elaboration.*
For example:
Icon showing images on the user's desktop;
Connects to (part of) the server/printer;
Operating system runs this access in the background (device drivers);

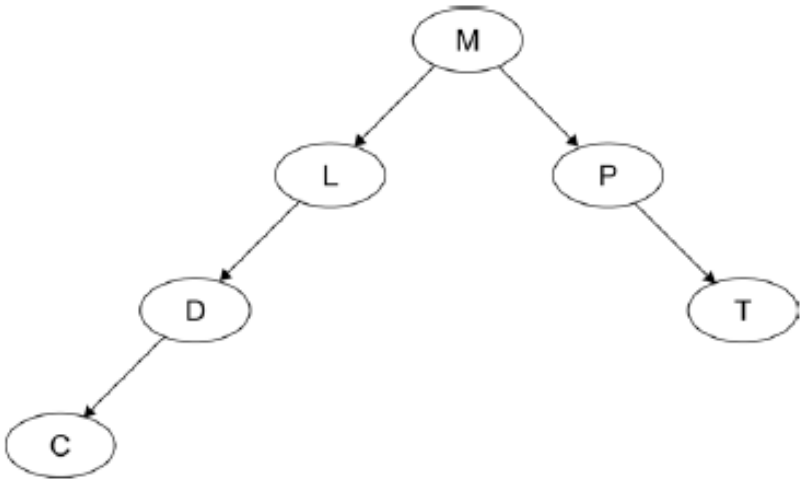
8. Consider the following binary tree.



(a) State the order that the nodes will be listed using the postorder tree traversal. [1]

CDLHTPM

(b) The node H is deleted so that the postorder traversal of the **remaining** nodes is preserved from part (a). Sketch the updated binary tree following this deletion. [2]



9. An insurance company holds a large database of information about its customers, including the date of their next payment.

Once a month the database is searched to compile the following lists:

- **list 1**: customers whose next payment date will be **within** the next 30 days
- **list 2**: customers whose payment date has passed by **more than** 14 days but **less than**, or **equal to**, 30 days
- **list 3**: customers whose payment date has passed by **more than** 30 days.

Records of customers who are in list 3 are flagged for deletion.

(a) Construct an algorithm to illustrate the monthly process described above. [6]

Award marks as follows, ***up to [6 marks max]****.*
Award ***[1 mark]*** *for looping through the database and accessing all records.*
Award ***[1 mark]*** *for correct calculation of date difference (eg = today – paymentDate OR paymentDate – today).*
Award ***[1 mark]*** *for each list correctly compiled,* ***x3*** *(correct conditional statements according to date difference used).*
Note: *Accept date difference not calculated/stated but assumed as today – paymentDate OR paymentDate – today.*
Award ***[1 mark]*** *for successive if/else but wrong conditions.*
Award ***[1 mark]*** *for flagging correct records for deletion (do not accept deleting the records).*

Example:

```
set CURRDATE to current date (as a day number)
set LIST1, LIST2 and LIST3 to empty
loop through all CUSTREC in DATABASE
     DUEPERIOD = CURRDATE - CUSTREC.PAYMENTDATE
     if DUEPERIOD > 30 then
          add CUSTREC to LIST3
          flag CUSTREC to delete
     else if DUEPERIOD > 14 then
          add CUSTREC to LIST2
     else if  DUEPERIOD < - 30 then
          add CUSTREC to LIST1
     end if
end loop
```

[6]

Note: *If candidates give their answer in flowchart form then credit them using the same marking points.*

After the lists have been compiled, the following messages are sent out to customers.

- A reminder is sent to customers in list 1.
- A warning that payments are more than 14 days overdue is sent to customers in list 2.
- cancellation of contract is sent to customers in list 3.

(b) Explain how the lists could be used to merge the data from the database with a word processor to create these messages automatically for sending either by post or by email. [4]

*Award up to **[4 marks max]**.*
(Using a mail merge facility);
Template for each type of reminder created in the word processor;
Lists created with customer ID;
Linked to customer details in database;
Appropriate details merged/inserted into template;

(c) Outline the consequences of data loss to customers and to the company. [2]

*Award **[1 mark]** for a consequence of data loss to customers and **[1 mark]** for a consequence of data loss to the insurance company.*

Example answer:
Customers would not be reminded when they needed to pay and some may overlook payment, hence not be insured;
The company could lose customers/ruin reputation;

(d) Describe **one** method that the company could use to prevent data loss. [3]

*Award marks as follows up to **[3 marks max]**.*
*Award **[1 mark]** for a suitable measure and **[2 marks]** for a description related to the insurance company.*

Example answers:
Mirror system;
All changes to the records made on two systems;
If one fails then the other holds all current data;

Off site backup;
Snapshots/backups made on a regular basis;
In the case of failure a dated/time stamped copy exists and the state up until then can be used to restore customer records;

10. Six lawyers and one secretary work together in the same building and are connected via a LAN to a central server. Each has their own workstation.

(a) Outline the concept of the Open Systems Interconnection (OSI) model in communication across a network. [3]

The OSI is a standardized system/model for network connection;
Consists of (7) layers;
Each dealing with specific parts of network communication;
For example, the physical layer which defines the physical connection;

Note: *Award* ***[1 mark]*** *for the purpose of any of the 7 layers.*
If candidate lists all 7 layers with no specific example award ***[2 marks]*** *and a further* ***[1 mark]*** *if the purpose of at least one layer is given.*

(b) Outline, with an example, the function of protocols. [3]

Award up to ***[3 marks max]****.*
Protocols are a set of rules;
To facilitate a process being carried out correctly;
(Used in each layer to ensure communication;)
For example (in the physical layer) the protocols could define the methods for opening and closing communication;

Note: *Do not accept examples which are not related to networks.*

The secretary deals with booking appointments for clients. New clients are given the first available appointment with any lawyer and returning clients are given the first available appointment with their usual lawyer.

A new customized computer package is bought to deal with appointment making.

(c) Identify the data that needs to be input by the secretary when someone asks for an appointment. [2]

Award up to ***[2 marks max]****.*
Name/ID;
Whether or not they are already a client;
If not a client, further details needed as input;

The data on appointments is held as one page for each day.

(d) Describe a suitable data structure to hold the data for one day. [3]

Two dimensional array;
With one column for each lawyer;
And one row per time slot;

Note: *Accept column or row for lawyer and vice versa for time slot.*

(e) Using the data structure you suggested in (d), outline the steps in a procedure to create an appointment for a client. [5]

*Award up to **[5 marks max]**. Accept answers given as an algorithm.*
Loading 'today' page (and from now onwards or accept "start with tomorrow");
If existing client, search appropriate lawyer/column only;
If not existing client, search time/row then lawyer/column;
Then allocate space if available;
If no space allocated, load next page and repeat until space found;
Then add client details to space;

11. In a small airport, the details of all flights due to arrive on a particular day are held in a collection, *FLIGHTS*. Each object in the collection contains the following information:

ID: unique flight number
PLACE: where the plane is coming from
DUE: the time it is scheduled to arrive
EXPECTED: the time it is expected to arrive (only if it is early or if it is delayed)
ARRIVED: the time of actual arrival.

EXPECTED and *ARRIVED* are blank at the beginning of the day and the collection is sorted in order of *DUE*.

A screen in the airport can display information on 20 planes at a time, which are held in a linked list.

(a) Describe the features of a linked list of 20 planes that have the above information. [3]

*Award **[1 mark]** for data, **[1 mark]** for pointers, **[1 mark]** for order.*
Example:
Each node would hold the data for one plane (ID, place, time due, time expected, landed);
Head pointer points to the first in the list;
Each subsequent pointer points to the next in the list and last node has null pointer;

All times are stored in the collection as the number of minutes since midnight. However, they are displayed on the screen in 24-hour format (for example, 10:58 is stored in the collection as 658).

(b) Construct an algorithm to convert the times held in the collection into hours and minutes needed for the 24-hour format displayed on the screen. [3]

*Award **[1 mark]** for calculating hours.*
*Award **[1 mark]** for calculating minutes.*
*Award **[1 mark]** for input and output/return.*

Example 1:

```
input CTIME // time held in the collection in minutes
     HOURS = CTIME div 60
     MINUTES = CTIME mod 60
output HOURS, MINUTES // time to be displayed on the screen
```

Example 2:

```
input CTIME // time held in the collection in minutes
HOURS = 0
MINUTES = CTIME
WHILE MINUTES>59
     MINUTES=MINUTES-60
     HOURS=HOURS+1
ENDWHILE
output HOURS, MINUTES // time to be displayed on the screen
```

Example 3:

```
Format24 (CTIME)
// method accepts time held in the collection in minutes
     HOURS = CTIME div 60
     MINUTES = CTIME mod 60
     return  HOURS + ":" + MINUTES
     // returns time to be displayed on the screen
end Format24
```

If a plane arrived more than 30 minutes ago it is removed from the linked list and the next one in the collection is added to the end of the list.

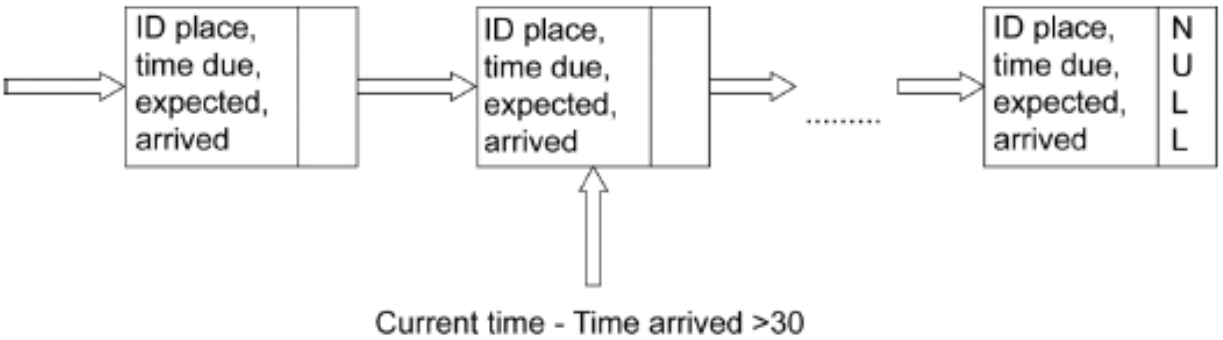
(c) With the aid of a diagram, explain how a plane which arrived more than 30 minutes ago could be removed from the linked list. [4]

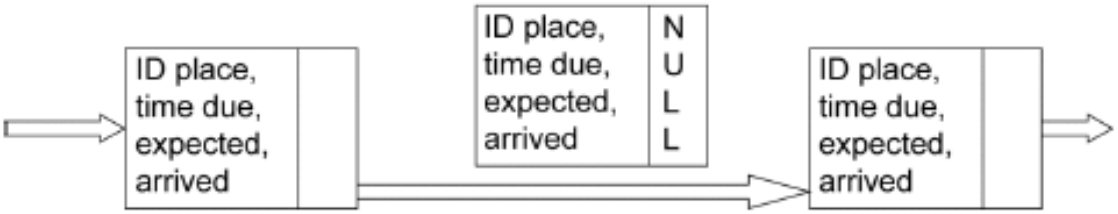
*Award marks as follows, up to **[4 marks max]**.*
*Award **[1 mark]** for a diagram and explanation showing access to each plane via pointers;*
*Award **[1 mark]** for comparison of current time with time arrived;*
*Award **[1 mark]** for correct change of pointer from plane deleted;*
*Award **[1 mark]** for correct change of pointer to next plane;*

***Note:** The plane to be deleted could be at the beginning of the list **OR** at the end of the list **OR** in the middle of the list; award third and fourth mark (change of pointers) depending on the position of the node shown in the candidates' diagram/explanation.*

For example:

```
PLANES accessed sequentially via pointers;
PLANE.ARRIVED checked against current time;
    if > 30 minutes;
        if pointer is head pointer;
            move head pointer to point to next PLANE;
        else if plane is last in list previous pointer points to NULL;
        else previous pointer changed to subsequent plane;
        pointer of deleted plane null;
```





(d) For the application described above, compare the use of a linked list with the use of a queue of objects. [5]

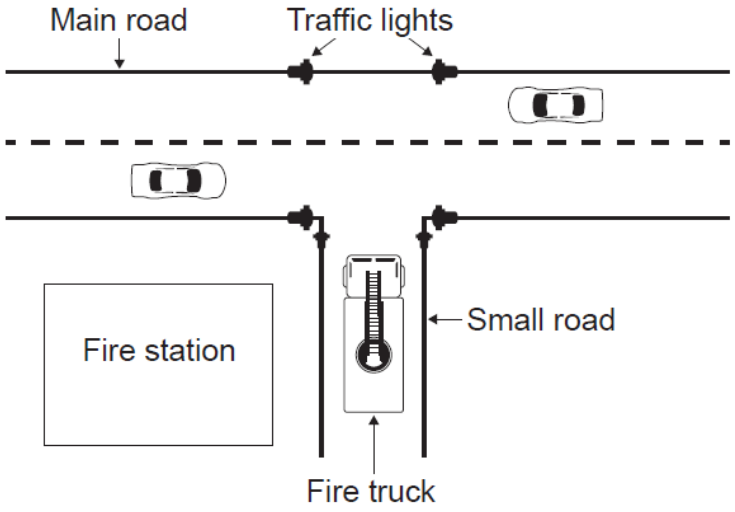
Award up to ***[5 marks max]***.
A queue would hold the elements in order of arrival;
And enqueue correctly to the end as required;

Dequeue would take planes from the top of the screen;
Which is not wanted as they arrive at different times;

Elements in a linked list could be removed from any position in the list;
Hence a linked list is better;

Searching for ID to amend will be equivalent;

12. In a town, a set of traffic lights control access from a small road, where a fire station is located, to a main road that has heavy traffic. In times of emergency, many vehicles from the fire station may need to leave the station at the same time. A system is put in place so that when a fire truck on the small road approaches the main road, the traffic lights switch to green (Go) on the small road and to red (Stop) on the main road.



(a) Outline the role of sensors and a microprocessor in controlling the traffic lights in this way. [4]

Sensors will be used to detect the approach of a vehicle from the minor road;
Likely to be touch/weight sensor embedded in the road;
Sensor input is converted from analog to digital;
To be processed and;
Signal sent to switch traffic lights;

(b) Suggest how the traffic lights can be changed back to their original state once there are no more fire trucks coming from the small road. [3]

*Award up to **[3 marks max]**.*
Continual feedback from sensor to processor;
A calculation based on number of vehicles/speed/*etc* or time taken for a vehicle to pass;
(Timer) resets if another vehicle is detected;
Once no input for a certain time traffic lights changed back;

These traffic lights are controlled by embedded systems at the point of use. It is proposed that they should be controlled from the same central computer as all the other traffic lights in the town.

(c) Discuss the advantages and disadvantages of running the town's traffic light system on one central computer with multiple inputs and outputs. [5]

*Award **[2 marks]** for advantages, **[2 marks]** for disadvantages and **[1 mark]** for weighing up.*

Disadvantages:
Central computer would have to cope with inputs from many places;
With differing priorities which could take time;
Connection failure possible from a particular point;
Computer failure puts all lights in the area out;
Cost of communication system/central control system;

Advantages:
More control over traffic flow at these points;
Lights can be adapted from distance to avoid traffic blocks;
Any problem appearing at one point is known immediately and can be dealt with;
Cheaper as no need for communication software/hardware/control centre;
Can react/change rules to changing levels of traffic flow;

Overall, it would be better to ... *(appropriate conclusion);*

A series of cameras are installed at each of the town's traffic lights. These cameras are connected to the central computer.

(d) Discuss the social implications of monitoring traffic in this way. [3]

*Award **[1 mark]** for an advantage outlined, **[1 mark]** for a disadvantage outlined and **[1 mark]** for discussing.*

For example:
Controlling the movement of vehicles and identifying people who speed should help to reduce accidents (as motorists know that they will be caught if driving dangerously);
This could also save lives;

Individual displacement is tracked;
Which can be seen as an infringement on personal liberty/a breach of privacy;
In some cases the information could be used unjustly against the individual (*eg* in times of political unrest);

It comes down to physical safety on the road against privacy/personal liberty;

13. Theo entered a maze (labyrinth) and tries to get to the centre. As soon as he arrived at the first possibility to turn right or left, he started recording each move on his phone so that he could find his way back to the start. He entered the moves as the direction he turned followed by the number of steps taken before the next turn. For example:

R3 , L5 , L10 , R6 , … , L4

which indicates "TURN **right**, STEP **3**", and then "TURN **left**, STEP **5**" etc.

An app on his phone stored the moves in a stack named STK, using 0 for "right" and 1 for "left".

The above moves were therefore stored as

0 , 3 , 1 , 5 , 1 , 10 , 0 , 6 , … , 1 , 4.

(a) Explain why a stack is a suitable structure to hold the data. [2]

Because items/moves are needed in reverse order;
To the order input;

OR

Because it is a LIFO (Last In First Out) data structure;
The items/moves pushed/placed onto the stack;
Will be popped off/taken from it in reverse order (to the order input);

Theo was successful in reaching the centre of the maze and now has to get back to the start.

(b) Construct an algorithm, using appropriate stack access methods, to output the moves needed to return from the centre to the first point where Theo started recording his moves. You can assume that he is **facing** the correct exit when he starts his return journey. [5]

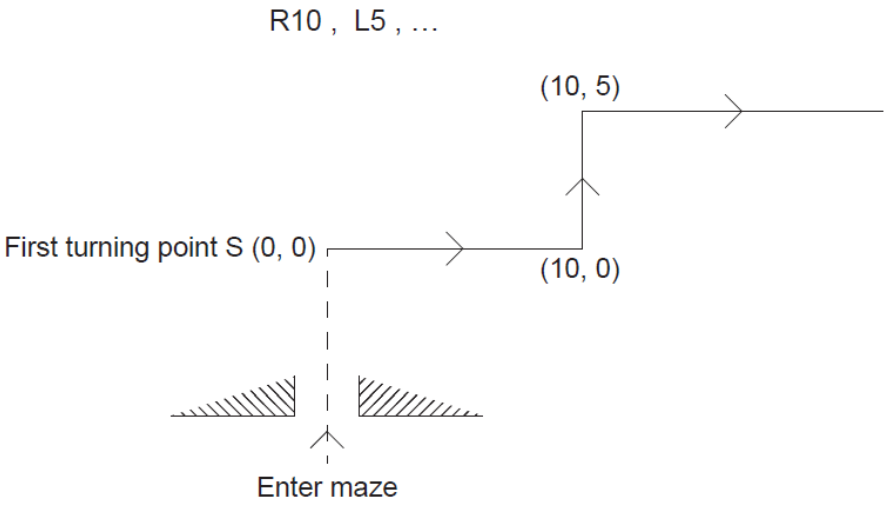
*Award marks as follows, up to **[5 marks max]**.*
*Award **[1 mark]** for checking for empty.*
*Award **[1 mark]** for popping STEP and TURN from stack.*
*Award **[1 mark]** for an output (STEP).*
*Award **[1 mark]** for correct if statement (checking popped TURN).*
*Award **[1 mark]** for an output (TURN).*

For example:

```
(let STK be the stack)
  while NOT STK.isEmpty() do
    STEP = STK.pop()
    output "Take " + STEP + " steps"
    TURN = STK.pop()
    if TURN = 0 then
      output "Turn left"
    else
      output "Turn right"
    end if
  end while
```

Another app on the phone gives Theo a visual representation of his path through a maze as a map. This app makes use of a procedure *MOVE()*, which outputs the coordinates of Theo's path through a maze, in reference to the point S, where he first turned right or left and which has coordinates (0, 0).

The diagram shows, for a **new maze**, the map from point S given the following moves:

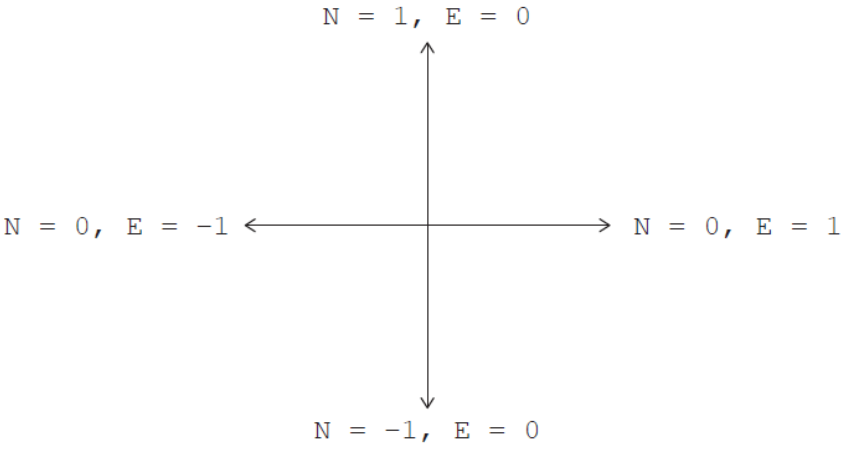


The third move is R8.

(c) State the coordinates on the map after this third move. [1]

(18, 5)

At each point, the direction in which Theo is facing is given by the variables *N* and *E*.
Note that before the very first turn is made, *N = 1* and *E = 0*.



The following table shows **part** of the trace of *MOVE()* according to the *TURN* and *STEP* values: R10 , L5 , R8 , R2 , L3 , R0.

The last move, with a *STEP* value of 0, indicates that there are no more moves and that the stack is empty.

(d) Complete the table by tracing the algorithm on the following page.

Move		Coordinates		Direction facing	
TURN	STEP	X	Y	N	E
		0	0	1	0
0	10	10	0	0	1
1	5	10	5	1	0
0	8				
0	2				
1	3				
0	0				

```
X, Y = 0                          // Initial coordinates
N = 1, E = 0                      // Direction facing at first turn
output (X, Y, N, E)               // Outputs starting point to the table
MOVE(X,Y,N,E)                     // Procedure to move on
     input (TURN, STEP)
     loop while STEP ≠ 0          // No more moves when STEP = 0
          if TURN = 0             // Right move
               X = X + N*STEP
               Y = Y - E*STEP
               if N = 0
                    N = -E
                    E = 0
               else
                    E = N
                    N = 0
               end if
          end if

          if TURN = 1             // Left move
               X = X - N*STEP
               Y = Y + E*STEP
               if N = 0
                    N = E
                    E = 0
               else
                    E = -N
                    N = 0
               end if
          end if
          output (X, Y, N, E)
          MOVE(X, Y, N, E)
     end loop
end MOVE
```

[6]

*Award marks as follows up to **[6 marks max]**.*
*Award **[1 mark]** for each correct pair X and Y (coordinates), **x3**.*
*Award **[1 mark]** for each correct change of direction facing (correct E and N), **x3**.*

Move		Coordinates		Direction facing	
TURN	STEP	X	Y	N	E
		0	0	1	0
0	10	10	0	0	1
1	5	10	5	1	0
0	8	**18**	**5**	**0**	**1**
0	2	**18**	3	**-1**	**0**
1	3	**21**	3	**0**	**1**
0	0				