IB Computer Science

System Development

Student_______________

Date: _________________

Directions:

Answer each part of the scenario completely. Be sure to address each "command term" and make your answer worthy of the points awarded for each part. Use complete sentences and answer on separate paper. Use your notes and Internet resources as needed to make sure you understand each concept that is addressed.

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]

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.
- A 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]

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

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

IB Computer Science

Student_______________

System Development

Date: _________________

Directions:

Answer each part of the scenario completely. Be sure to address each "command term" and make your answer worthy of the points awarded for each part. Use complete sentences and answer on separate paper. Use your notes and Internet resources as needed to make sure you understand each concept that is addressed.

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]

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]

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]

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

IB Computer Science

System Development

Student_______________

Date: _________________

Directions:

Answer each part of the scenario completely. Be sure to address each "command term" and make your answer worthy of the points awarded for each part. Use complete sentences and answer on separate paper. Use your notes and Internet resources as needed to make sure you understand each concept that is addressed.

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]

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]

IB Computer Science

System Development

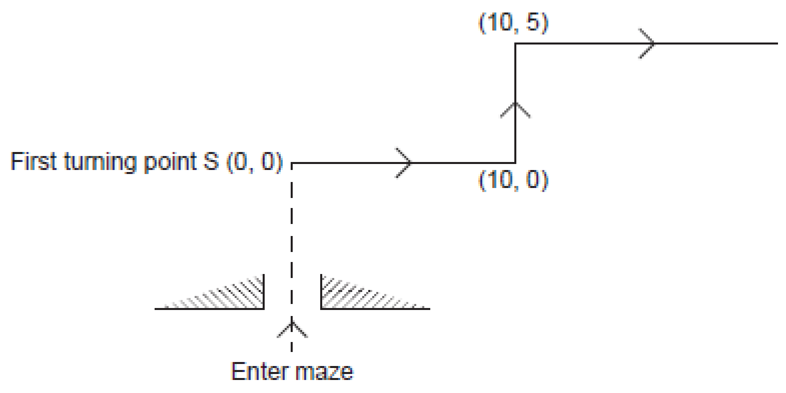
Student_______________

Date: _________________

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:

R10 , L5 , ...



The third move is R8.

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

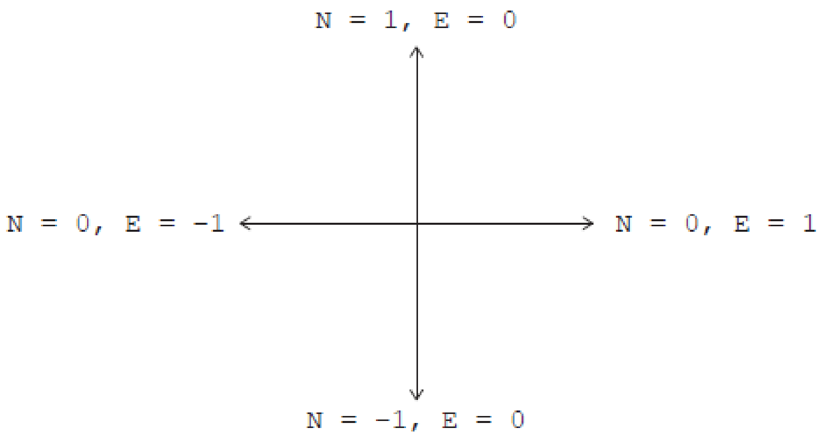
IB Computer Science

Student________________

System Development

Date: __________________

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.

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				

D) Trace the algorithm on the following page to complete the table above [6]

IB Computer Science

Student_______________

System Development

Date: _________________

```
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
```