M18/4/COMSC/HP1/ENG/TZ0/XX/M

Markscheme

May 2018

Computer science

Higher level

Paper 1

14 pages

Section A

1. (a) *Award* ***[2]*** *for one method that is suitable for the given scenario.*

Answers may include:
<u>Interviews</u> could be held (with the <u>librarian/stakeholders)</u>;
To establish the functions required by the system;

<u>Direct observation</u> could be made of the users/<u>students</u> using the present system;
To gain an insight on how the library is being used; **[2 max]**

(b) Allows stakeholders to gain an idea of how the system would be/work/look;
so they can provide feedback / suggest improvements; **[2]**

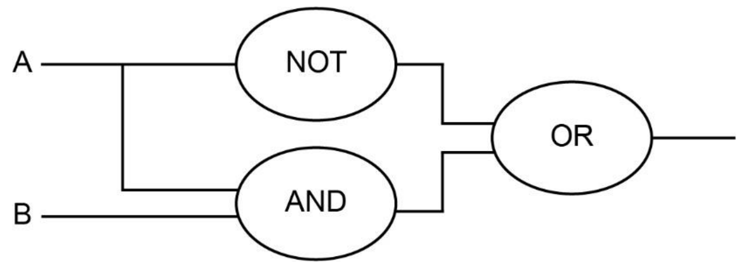
2. Holds (a copy of) the contents of the memory;
Which are to be transferred from/to memory to/from other CPU components;
Allowing the processor and memory to act independently/processor not affected by differences in the speed of operation; **[2 max]**

3. 2 hex = 4 + 4 = 8 bits;
hence 2^8/16^2/256 colours;
Note: *Allow* ***[2]*** *for 256 colours with no workings and also ONLY 256* **[2]**

4. *Award* ***[3]*** *as follows:*
The function of the memory management **[1]**.
An example if its use **[1]**.
Reason for the importance **[1]**.

<u>Example:</u>
Allocates and de-allocates memory/ assigns blocks of memory to programs;
Ensures a program has sufficient memory to run/manages virtual memory if needed;
To avoid overwriting /clashing of programs/optimise system performance/maximise memory usage **[3]**

5. *Award* ***[3]*** *as follows:*
Clearly a logic diagram with 2 inputs, 1 output and 3 logic gates **[1]**;
OR gate has two inputs one of which is NOT A **[1]** and the other is A AND B **[1]**



[3]

6. 47, 23, 11
*Award **[1]** per output.*
***Note:** Allow FT if either of the first two outputs are wrong.* **[3]**

7. Process/method/ procedure/ subroutine/ function/ algorithm that calls itself; **[1]**

8. *Award **[2].***
FIFO data structure;
Items can be added only to one end (rear/tail) and removed from the other end (front/head); **[2]**

9. *Award **[3]**.*
*Award **[1]** for size comparison.*
*Award **[1]** for access comparison.*
*Award **[1]** for relation to the given scenario.*

<u>Example</u>:
Size/length of a static array is predetermined/fixed / size of a linked list can be changed / size of a linked list depends only on memory available;
Linked list can be expanded to suit the daily sales;
Each item in a static array can be directly accessed whilst access to the items in a linked list is sequential;
And searching for the given item in the linked list is slower /an item in the linked list cannot be searched for using binary search; **[3]**

10. *Award **[2 max].***
An interrupt is a signal to the CPU sent by hardware or software;
The function of an interrupt is to alert the CPU to suspend execution of the current program;
And to transfer the control to the interrupt handler (which saves the state of the current program to the interrupt stack, services the interrupt and resumes the normal CPU activity); **[2]**

Section B

11. (a) Award marks for a response which indicates the logical steps that have to be followed.

*Award **[5 max]** as follows*:
Iterate through the appointments file;
Check for correct day;
Repeat for each appointment on that day;
Using the patient ID for that appointment;
Iterate through the patients file until record for that ID found;
Retrieve phone number and send out SMS;

<u>Example:</u>

```
loop for every appointment in the appointments file
    if the appointment is for the correct day
        Store patient ID
            loop until found in the patients file
                if patient ID (patients file) = patient ID (appointments file)
                    get phone number from patient’s file to send SMS
                end if
            end loop
    end if
end loop
```

***Note:** Candidates are not requested to construct the (algorithm in) pseudocode.* **[5 max]**

(b) *Award **[1]** for method and **[1]** for description – only accept TWO methods.*
*Mark as **[2]** and **[2]**.*

Backup;
Data files on a regular basis;

Printed copies;
Printouts can be kept of transactions;

Transaction Log file;
Written for each transaction can be used to restore;

Accept any reasonable methods described including second server and cloud use. **[4 max]**

(c) *There are 2 possible issues here: who has what level of access to the data in the hospital and whether storing in the cloud is safer than storing locally.*

For each of these 2 issues award ***[3 max]*** *as follows:*
Award ***[1]*** *for identifying the issue.*
Award ***[1]*** *for some valid development of the issue.*
Award ***[1]*** *for a suitable discussion.*

Example answers ***could*** *include reference to the following but this is not an exclusive list. Award marks for any two reasonable issues discussed – one of which is access and the other security.*

Official access to the data [3 max]:
Access to this sensitive data must be restricted.
Only those directly concerned can be able to access it.
Even less people should be able to edit it.
Therefore access levels should be set up, with strong levels of authentication.
Physical access to servers should be controlled if using the local system;

Data security [3 max]:
Is the data safer stored locally or on the cloud?
Cloud service providers are professionals – they should have stronger security than a hospital system.
What track record / reputation does the cloud service provider has?
If patient data could be sold/inspected, then both the patient and hospital could suffer serious consequences.
Is the cloud governed by appropriate privacy laws?
Is it located internationally or is it governed by the laws of the country in question?
Could be intercepted in transmission; **[6 max]**

12. (a) Processes/task are carried out simultaneously/at the same time; **[1]**

(b) Install connectors on wall of server room AND Install connector on wall of new office space;

OR

Test the cabling AND Connect the new computers with the cabling; **[1 max]**

(c) *Any pair of tasks that are NOT a correct answer to part (b)*; **[1 max]**

(d) *Award* ***[1]*** *for stating an advantage,* ***[1]*** *for expansion/example in context.*
Mark as ***[2]*** *and* ***[2]***

Answers may include:
Use on the move;
More versatile staff encouraged to collaborate *etc.*;

Allows BYOD:
Which could lead to greater productivity (as familiarity with device);

No extra equipment is needed for expansion after initial set-up;
Which will save the company time and money;

Reduces wiring;
Therefore improved safety for employees; **[4 max]**

(e) *Mark as* ***[2]*** *and* ***[2]****.*

The data can be intercepted as it goes through "the air";
Can be resolved by strong encryption/protocols;
WPA-2 / a description of WPA-2;
Use of trusted MAC addresses;
Regular changes of router password; **[2 max]**

BYOD issues leading to insecure devices;
Clear company policy regarding use;
Use of sand-box;
Only approved devices allowed;
MAC addresses – only adding clean and tested devices brought in by staff;
Installation of MDM services;
Authentication (user ID + password on all devices including BYOD);
Security features added by company; **[2 max]** **[4 max]**

(f) A VPN/ tunneling allows the employee's device to appear to be part of / a node of the internal company network;
Thus affording him/her full access to the network resources;

Data that passes through a VPN can be encrypted;
So any unauthorized access will not be able to understand the data;

Tunnelling allows the company's own protocols to be used/IPsec/TSL ensure security;
Even though the data is passing over an outside network;

Multiple exit nodes / hidden IP addresses/encrypted connections;
Make it hard to distinguish where the data was generated; **[4 max]**

13. (a) *Award **[5 max]** as follows:*

Example answer 1:

```
searchName(NAME, STACK)
  DEPTH = 0
  NAMEDEPTH = -1
  while not STACK.isEmpty() do
    X = STACK.pop()
    if X == NAME
      then NAMEDEPTH=DEPTH
    DEPTH = DEPTH + 1
  end while
  if NAMEDEPTH == -1
    then output(NAME + 'not found')
    else output(NAMEDEPTH)
  endif
end searchName
```

*Award **[1]** for initialization.*
*Award **[1]** for a correct while/until loop.*
*Award **[1]** for correct comparison and assignment.*
*Award **[1]** for updating and outputting correct position of `NAME` on the stack(`NAMEDEPTH`)*
*Award **[1]** for use of stack methods (`pop()` and `isEmpty()`)* **[5]**

Example answer 2:

```
NAME=input()
CSTK = STACK.copy() // must make a real copy,
//any attempt to make a copy of stack is acceptable
DEPTH = 0
SEARCHING = true
while SEARCHING and not CSTK.isEmpty() do
  if NAME == CSTK.pop()
  then
    SEARCHING = false
  else
    DEPTH = DEPTH + 1
  end if
endwhile
if not SEARCHING
  then output(DEPTH)
  else output NAME + " not found"
end if
```

*Award **[1]** for initialization.*
*Award **[1]** for a correct while/until loop (even if flag is missing).*
*Award **[1]** for correct comparison and assignment.*
*Award **[1]** for incrementing and outputting correct position of `NAME (DEPTH).`*
*Award **[1]** for use of stack methods (`pop()` and `isEmpty()`).* **[5]**

(b) *Award **[3 max]**.*

Example answer 1 (time efficiency):
The data in the binary search tree(BST) is ordered;
Such that as each node is checked for the item, half of the remaining nodes are ignored;
But each element in the stack has to be checked;
Which, for large data sets, may be inefficient compared to the BST;

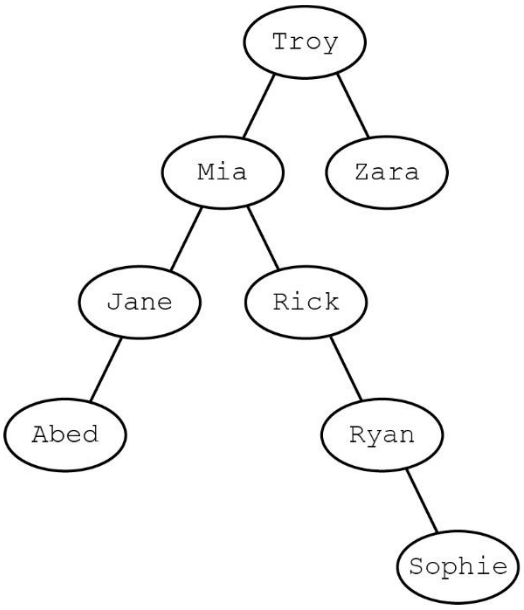
Example answer 2 (memory efficiency):
Each element on the stack has to be popped off/removed from the stack to be checked for the searched item;
When the item is found the stack will be empty/stack contents will be changed;
When an element in the BST is checked for the item it is not removed from the BST;
So there is no need to create an additional copy of the BST (but a real copy of the stack must be created which for large data sets may be inefficient); **[3 max]**

(c) *Award **[4 max]**.*

Example answer:
The (next) stack item is (placed into a new node and) compared (alphabetically) to the root;
If the root is empty it would be placed here (and this recursive procedure terminates);
Else, depending upon the comparison, it would look to the node to the left or right and the (recursive) procedure calls itself again (with the new root);
If it is lower than the root, then the left child of the root becomes the new root;
If it is higher than the root, then the right child of the root becomes the new root;

***Note**: Marks should also be awarded for an algorithm constructed in pseudocode.* **[4]**

(d) *Award **[3]**.*



Award marks as follows:
Clearly a binary tree;
Correct root;
All values correct; **[3]**

14. (a) *Award **[1]** for identifying the hardware component and **[1]** for describing its purpose within the control system.*
*Mark as **[2]** and **[2]**.*

Sensor;
To detect the presence/absence of the light;

Microprocessor;
To carry out any processing;

Output transducer / Actuator / Activator;
To turn the lights on or off; **[4 max]**

(b) The output value is (continuously) compared to the desired value;
To produce an error value/difference between observed and measured;
The controller uses the error value/difference between observed and measured;
To determine the new input to the system; **[3 max]**

(c) *Award **[5 max]** for the following:*
***[1]** the system is turned on (when the light intensity is below a given value);*
***[1]** for light loop/while input light intensity is below the given value the system continues being on;*
***[1]** for comparison and activating watering / checking the water content of the flower beds and turning on water if the water content is not in a given range;*
***[1]** for water loop / watering until maximum value;*
***[1]** for turning off water when the water content of the flower beds is filled to maximum;*
***[1]** for turning off system when the light intensity increases the given value;*

***Note:** Candidates are not requested to construct an algorithm in pseudocode.*

Example answer:

```
turn on system
loop while INPUT_LIGHT < MAX_LIGHT
    if INPUT_WATER < MIN_WATER
        turn on water
        loop while INPUT < MAX_WATER
            // system continues being on
        end loop
        turn off water
    end if
end loop
turn off system
```

where
INPUT_LIGHT // inputted value of light intensity
MAX_LIGHT // maximum value of light intensity
INPUT_WATER // inputted value of water content
MIN_WATER // minimum value of water content
MAX_WATER // maximum value of water content **[5]**

15. (a) *Award **[4]** as follows.*

Example answer 1:
Assumes that an array of sufficient size is declared (`2D_ARRAY[][]`) and `PASSENGERS` contains valid data for all 7 days in each of 30 weeks.

```
PASSENGERS.resetNext()
N = 0
loop while PASSENGERS.hasNext()
        //accept not PASSENGERS.isEMPTY()
        //accept N<PASSENGERS.length()
   WEEK = N div 7
   DAY = N mod 7
   2D_ARRAY[WEEK][DAY] = PASSENGERS.getData() // PASSENGERS.getNext()
   N=N+1
end loop
```

*Award **[1]** for while loop.*
*Award **[1]** for correct calculation of row index(WEEK)*
*Award **[1]** for correct calculation of column index(DAY)*
*Award **[1]** for correct array subscripts*
*Award **[1]** for correct use of collection methods*

Example answer 2:
Assumes that an array of a sufficient size is declared (`2D_ARRAY[][]`) and `PASSENGERS` contain valid data for all 7 days in each of 30 weeks)

```
PASSENGERS.resetNext()
loop for WEEK from 0 to 29
   loop for DAY from 0 to 6
      2D_ARRAY[WEEK][DAY] = PASSENGERS.getNext()//PASSENGERS.getData()
   end loop
end loop
```

*Award **[1]** for nested loops.*
*Award **[1]** for loops that generate the right output (correctly read `PASSENGERS` into `2D_ARRAY`).*
*Award **[1]** for assignment into correct array location.*
*Award **[1]** for correct use of collection methods.* **[4]**

(b)
```
total (N)
// if N (column number) is not passed as an input parameter then
// assignment must appear in pseudocode, for example, N = input()
   SUM = 0
   loop for WEEK from 0 to 29
      SUM = SUM + 2D_ARRAY[WEEK][N]
   end loop
   return SUM
end total
```

*Award **[1]** for assigning the input to a variable / passing the value of `N`*
*Award **[1]** for initializing and returning/outputting `SUM`*
*Award **[1]** for correct loop*
*Award **[1]** for correct addition of each term to `SUM` (**Note**: the array position must be completely correct with the input column and the varied row).* **[4]**

(c)
```
MAXAV = 0
MAXDAY = ""
loop for DAY from 0 to 6
     if total(DAY)/30 > MAXAV
       MAXAV = total(DAY)/30
       MAXDAY = convert(DAY)
     end if
  end loop
output MAXAV,MAXDAY
```

Award ***[1]*** *for correct loop.*
Award ***[1]*** *for use of method total() when calculating the average number of passengers.*
Award ***[1]*** *for initializing, updating and outputting/returning `MAXAV` and `MAXDAY`.* **[3]**

(d)
```
salesCalculate(2D_ARRAY, A, B, X, Y)
// A,B are row & column of first day,
// X,Y are row & column of last day.

TOTAL = 0

loop for COL= B to 6
  if COL < 5 then
    TOTAL = TOTAL + 2D_ARRAY[A][COL] * FEES[0]
  else
    TOTAL = TOTAL + 2D_ARRAY[A][COL] * FEES[1]
  end if
end loop
        //calculates total for week A,
        //days are in the range from B to 6
        //weekdays are days 0-4, and weekend 5,6

loop for ROW = A + 1 to X - 1
  loop for COL= 0 to 6
    if COL < 5 then
      TOTAL = TOTAL + 2D_ARRAY[ROW][COL]*FEES[0]
    else
      TOTAL = TOTAL + 2D_ARRAY[ROW][COL]*FEES[1]
    end if
  end loop
end loop
        //calculates total for all weeks from A+1 to X-1,
        //days are in the range from 0 to 6

loop for COL= 0 to Y
  if COL < 5 then
    TOTAL = TOTAL + 2D_ARRAY[X][COL]*FEES[0]
  else
    TOTAL = TOTAL + 2D_ARRAY[X][COL]*FEES[1]
  end if
end loop
        //calculates total for week X,
        //days are in the range from 0 to Y

output TOTAL
end salesCalculate
```

*Award **[7 max]**.*
*Award **[1]** for initializing, updating and outputting TOTAL.*
*Award **[1]** for nested loops.*
*Award **[1]** for correct calculation of `TOTAL` in row/week A (column subscripts/days/ in `2D_ARRAY` must be in range B to 6).*
*Award **[1]** for correct calculation of `TOTAL` in row/week X (column subscripts/days/ in `2D_ARRAY` must be in range 0 to Y).*
*Award **[1]** for correct calculation of `TOTAL` in rows/weeks from A + 1 to X – 1 (column subscripts/days/ in `2D_ARRAY` must be in range 0 to 6).*
*Award **[1]** if evident that the number of days to be included in calculating total amount is different for weeks A + 1 to X – 1, week A and X (three different cases).*
*Award **[2]**, **[1]** for checking for weekdays / weekend and **[1]** for using correct subscript in FEE (0 for weekdays, 1 for weekend).* **[7 max]**