Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

November 2022

Computer science

Higher level

Paper 1

17 pages

Subject details: Computer science HL paper 1 markscheme

Mark allocation

Section A: Candidates are required to answer **all** questions. Total 25 marks.
Section B: Candidates are required to answer **all** questions. Total 75 marks.
Maximum total = 100 marks.

General

A markscheme often has more specific points worthy of a mark than the total allows. This is intentional. Do not award more than the maximum marks allowed for that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

Each statement worth one point has a separate line and the end is signified by means of a semi-colon (;).

An alternative answer or wording is indicated in the markscheme by a "/"; either wording can be accepted.

Words in (…) in the markscheme are not necessary to gain the mark.

If the candidate's answer has the same meaning or can be clearly interpreted as being the same as that in the markscheme then award the mark.

Mark positively. Give candidates credit for what they have achieved and for what they have got correct, rather than penalizing them for what they have not achieved or what they have got wrong.

Remember that many candidates are writing in a second language; be forgiving of minor linguistic slips. In this subject effective communication is more important than grammatical accuracy.

Occasionally, a part of a question may require a calculation whose answer is required for subsequent parts. If an error is made in the first part then it should be penalized. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with "**FT**".

General guidance

Issue	Guidance
Answering more than the quantity of responses prescribed in the questions	In the case of an "identify" question, read all answers and mark positively up to the maximum marks. Disregard incorrect answers. In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications. In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation *etc*.

Section A

1. *Award **[2]** max.*
A human-readable form of the program should be translated into executable binary code / machine language code;
which is the only language computer hardware can understand;

2. (a) *Award **[2]** max*
Prototyping approach allows the end users (employees, management, owner) to visualize how the final solution might look;
So, they can provide feedback to the system analysis team;

Gives the design team the chance to explore different possibilities;
Before choosing the best one / most likely one;

Feedback from end users;
Saves wasting time at a later stage;

***Note**: Reward other suitable answers.*

*Award **[1]** for stating an advantage and **[1]** for an expansion.*

(b) *Award **[2]** max.*
Software developer works on a rather simple project / in a small company;
No end-users so no need for feedback / to explore different possibilities;

The software developer is the only end-user;
No need to waste time and money on a prototype / to view how the solution might look;

When designing very large/resources intensive systems;
A prototype may be too expensive/ cost inefficient;

In a case where trial/error is a dangerous approach/can be lethal/harmful for human lives/unethical;
For example, systems that overlook nuclear power plants/ surgical equipment, etc.

3. *Award **[2]** max.*
The XOR operation takes two Boolean operands and returns true (if and only) if the operands are different;
it returns false if the two operands have the same value;

***Note:** If defined by drawing the truth table; then award 1 mark for all four combinations of two inputs and 1 mark for the correct output column.*

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

4. (a) *Award **[2]** max.*
the video/images could be viewed by anyone;
which may not be desirable since some people may not wish to have themselves regarded by everyone;

anybody could download the images/video;
and they can be manipulated by outsiders;

Posting images of other people at a private function requires their permission;
did she get permission from her friends? ;
that they were fine with her posting and share their pictures with everyone;

(b) *Award **[2]** max.*
there could be problems if the person at the event does not have the facility to see the video/images;
because of, for example, inadequate internet connection / browser / video player;

space on web server could cause problems;
if the size of files exceeds the allocation on server;

5. (a) *Award **[1]** max.*
000000000000000011111111;

(b) *Award **[1]** max*
2^{24} / $(2^8)^3$ / 256^3;
16.8 million / 16,777,216;

6. *Award **[1]** max.*
Variables define areas in memory in which values (data) are stored;
Variables are unique identifiers that retrieve values from the memory addresses at which they are stored;
Variables are the names one gives to computer memory locations which are used to store values in a computer program;

7. *Award **[3]** max.*
Commonly used instructions from a program are stored in cache memory to be accessed by processor (faster access than from RAM);
Increasing the size of cache memory means that more instructions can be quickly found/accessed by the processor;
therefore, program execution is made faster;

8. (a) *Award **[1]** max.*
A data structure in which items are added to the tail/rear (one end) and removed from the head/front (another end) / first in first out (FIFO) data structure;

(b) *Award **[2]** max.*
Print queue;
Keyboard queue;
Accept any other job queue (program is running!);

9. *Award **[4]** max.*
***Note:** The work out may be differently presented.*
*1 mark for each name outputted in the correct order**, x4**.*

The first call mystery (NAMES,3)
The second call mystery (NAMES,2)
The third call mystery (NAMES,1)
The fourth call mystery (NAMES,0)
N=0 **output: Kim**
End of call mystery (NAMES,0)
N=1 **output: *Eli***
End of call mystery (NAMES,1)
N=2 **output: Murali**
End of call mystery (NAMES,2)
N=3 **output: Zixan**
End of call mystery (NAMES,3)

Note:
*Award **[2]** marks in case that **only** output*

Kim, Eli, Murali, Zixan

is shown. Award 1 mark for 4 names outputted and 1 mark for the correct order of names.

Section B

10. (a) *Award **[1]** max.*
(Wireless) router;
Access points;
(Wireless) bridge or repeater;
(Wireless) controller;
Wireless adaptors/ Network interface cards (NICs) (that operate within the employees' computers);

(b) *Award **[4]** max.*
The reliability of wireless depends on the strength of the wireless signal;
Depends on distance from router;
depends on the topology/shape of the surroundings;
a wireless LAN has slower data transfer;

whilst
Ethernet is more reliable as the strength of the signal does not depend on the distance from the router/ wired LAN support longer distances;
wired is immune to interference;
there is no issue with the topology/shape of the surrounding;
but the Ethernet cable may be cut/broken affecting reliability;

(c) *Award **[2]** max.*
wired network is only accessible with a physical cable connection and in wireless networks signals are broadcasted outside of the building;
leaving it open to the public and potential hackers/ it is easy to intercept transmissions;

a wireless network is more open to misuse;
as network administrator cannot directly monitor a specific machine;

(d) *Award **[4]** max.*
Authentication;
Nobody outside the VPN should be able to use the VPN and affect the security of the VPN, so only user name and password authentication is not enough;
Strong user VPN authentication uses various methods, including certificates, one-time passwords and software tokens;

Encryption;
If data intercepted, it will not be readable;
encrypting each encapsulated data packet's content with an encryption key (the key is shared only between the VPN's server and clients);

Tunnelling software;
A VPN hides a user's data by encrypting it with a tunnel created between the user's device and the VPN's web server (The user then takes on the IP address of the web server (rather than their true IP), and this leads to one advantage of a VPN (namely that a user can appear to be in a different geographic location than they are actually located in);

Multiple exit nodes;
Makes it hard to distinguish where the data sent is originated (protecting privacy);

*Mark as **[2]** and **[2]***
*Award **[1]** for a feature and **[1]** for a reasonable expansion, x2.*

(e) *Award **[4]** max.*
Data is organized in specially formatted units (data packets);
Each data packet contains: data, address of the sender, address of the receiver, size of packet, sequence number, control codes, etc.;
which are routed from source to destination using network switches and routers;
network switches and routers determine how best to transfer the packet between a number of intermediate devices (routers and switches) on the path to its destination (rather than flowing directly over a single wire on the path to its destination);
data packets are reassembled at destination;

11. (a) (i) *Award **[1]** max.*
Data migration is the process of transporting/transferring data between computers / storage devices;
Data migration is importing the data/database(s) (created in Africa) into the new database(created in Europe);

(ii) *Award **[4]** max.*
Different languages used;
Which leads to translation/communication issues;

Different character sets used;
Which could lead to inconsistent/incorrect results;

Incompatible file formats;
Which could lead to incomplete or incorrect data transfer / data loss;

Different file structures / different employee records;
Which will result in a mismatch of data/ incorrect data;

New validation rules differ to the old validation rules (used in the Africa headquarters);
Which could lead to inconsistent/incorrect results;

*Mark as **[2]** and **[2]***
*Award **[1]** for an issue and **[1]** for a reasonable expansion, x2.*

(b) *Award **[4]** max.*
Issues with personnel;
such as redundancy/retraining/employee not willing to move to Europe, etc.;

Changeover decisions;
such as direct changeover/ parallel running etc;

Time needed;
for merging/aligning the two systems;

Costs involved;
in the aligning of the two systems/ in the changeover period of time/ for buying new software or hardware/ cost for additional working hours/ cost of additional staff;

Testing;
Of the new systems with the new data;

Data entry;
If migration not possible/ cost and time involved;

*Mark as **[2]** and **[2]***
*Award **[1]** for identifying an aspect that could arise and **[1]** for an expansion, x2.*

(c) *Award **[3]** max.*
Applications in the legacy/outdated system do not have upgrades;
Documentation of the legacy system might be missing/incomplete/unreliable;
Data cannot be converted to newer formats;
Legacy systems are typically large and complex systems/ mainframes, so programs might be disorganized/hard to understand/expensive to change;
It may be difficult to find system engineers/ employees/ programmers familiar with the old programming languages/ old operating systems to maintain the system;

(d) *Award **[3]** max.*
Two systems will be run simultaneously so that operations are not disrupted, this is a costly operation;
Because both systems and all their resources need to be maintained/run at the same time;
Data migration has many risks/is a complex process so the costs to implement it correctly are high because of additional costs due to delays /lost time/data entry;
The decision on when and whether the new system is ready to remove the old one depends on the testing and validation of the new system, which includes some additional costs associated with the testing/validation of the system;

***Note**: Reward other suitable responses.*

12. (a) *Award **[4]** max.*
Heating / air conditioning controlled;
help lower the energy bills / increases the comfort level of the occupants;

Monitoring hot water system;
temperature and pump control monitoring via the building monitoring system allows for a proper functioning of hot water distribution through the facility;

Security system;
Locking/unlocking doors for the authorised or unauthorised employees/ windows open-close;

Sprinkler system;
Monitored for fire safety;

Monitoring environment;
to know whether we can take advantage of it;
Accept examples, such as, lights can be turned off if outdoor luminosity is high enough for indoor working;

Electrical system;
the consumed electrical power and the state of main electrical switches can be controlled/monitored;

Monitoring/controlling pressure/humidity;
and if their value exceeds defined limits, an alarm is generated;

*Mark as **[2]** and **[2]***
***Note**: Reward other suitable answers.*

(b) *Award **[4]** max.*
electromagnetic sensors detect if a door is opened/closed;
(*Accept other appropriate type of sensors!*
For example, proximity sensor detects when a person comes close to the door, movement sensor detects people approaching the door.)
the read/detected sensor data is transmitted to processor;
processor determines whether any action should be taken (if a change in the state is detected-action needed);
then processor sends the signal (about which door to close/open/how long to keep it open) to the output transducer/ actuator;
which opens/closes the door:
the whole process is continually repeated;

(c) (i) *Award **[2]** max.*
Managing peripherals;
Managing memory;
Resource allocation;
***Note:** Reward other correct responses.*

(ii) *Award **[2]** max.*
Interrupt;
An input device sends a signal which causes the processor and the OS to stop the current operation and determine what to do next;

OR

Polling;
All input devices are continuously checked by the processor to see whether they want to communicate;

(d) *Award **[3]** max.*
Award marks for the benefits such as energy savings, productivity gains, reducing heating costs, increase the productivity of the facility staff, improve building operations, support sustainability efforts; enhance decision-making across the organization, etc.

energy savings;
the use of optimal start/stop;
which allows the building automation system to learn, for example, when it should bring the air conditioning system online (for a particular zone in the building);

Improved asset performance;
which reduces energy use / optimizes how space is used;
and minimizes the environmental impact of buildings;

13. (a) (i) *Award **[5]** max*
*Award **[1]*** for repeating the following steps until no swaps is required (the array is sorted);
*Award **[1]*** for looping through the array SCORES;
*Award **[1]*** for comparing each pair of adjacent elements in the array SCORES;
*Award **[1]*** for swapping them if they are in the wrong order;
*Award **[1]*** for also swapping the corresponding elements in the array NAMES;

(ii) *Award **[1]** max.*
Selection sort;
Insertion sort:
Quick sort;
Merge sort;
Shell sort;

(b) *Award **[3]** max.*
Components data / two data fields: name and score;
pointer to the left child;
pointer to the right child;

(c) *Award **[6]** max.*
*Award **[1]*** for *a loop (to input all the data)*
*Award **[1]*** for *placing the inputted data into a newly created node*
*Award **[1]*** for *setting the root node to the new node(if the tree is empty/ no root node)*
*Award **[1]*** for *start searching from the root node*
*Award **[1]*** for *searching for the empty location in the left subtree if the input score is higher than the current value*
*Award **[1]*** for *inserting the new node in the left subtree when the proper location is found*
*Award **[1]*** for *searching for the empty location in the right subtree if the input score is **lower** than the current value*
*Award **[1]*** for *inserting the new node in the right subtree when the proper location is found*

Example 1:
Repeat the following steps (until all the names and scores are inputted);
input the values (name and score of one student), create a new node and place the input values into the newly created node;
If the tree is empty, set the root to a new node;
Else (if the tree is not empty)
repeat the following until a leaf node is reached;
If a new node is smaller than or equal to the node (here it is a root node) move to its right child;
If a new node is larger than the node, move to its left child:
insert the new node (as a right or left child of the leaf node based on node is less or greater than the leaf node);

Example 2:
Repeat the following steps (until all the names and scores are inputted);
Create a new node (for example, named ITEM); input the values (name and score of one student) and place them into the new node (ITEM) (and set its left and right to NULL);
If there isn't a root node, set the root node to ITEM;
If there is a root node, start searching from the root node (to find ITEM's proper location),
if the score in ITEM is greater than the score in the root search for the empty location in the left subtree;
otherwise (if the score in ITEM is lesser than the score in the root), search for the empty location in the right subtree;
insert ITEM into empty location;

Example 3:
While the end of the input list is not reached do the following;
input the values (name and score of one student), create a new node and place the input values into the newly created node;
if there is no root node, set the root to the new node;
if there is a root node
start from the root node and if the score in the node to insert is less than the root, go to right child;
otherwise go to the left child of the root;
continue this process (Note: each node is a root for some subtree) until a null pointer (or leaf node) is found (where it is not possible to go any further);
when the leaf node is found, insert the node as a right or left child of the leaf node (based on node is less or greater than the leaf node);

14. (a) *Award **[2]** max.*

*Award **[1]*** for *checking that the remainder in integer division is equal to zero*
*Award **[1]*** for *also checking that X is not equal to Y.*

Example:

```
(Y mod X == 0) and (X != Y)
```

***Note:** Accept `%` instead of `mod`. Accept `&&` instead of `and`.*
*Accept other correct expressions such as (X ≠ Y) and (trunc(Y/X)*X = Y).*

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

*Award **[1]*** for *initializing SUM and correct increment*
*Award **[1]*** for *the correct loop*
*Award **[1]*** for *checking the remainder in integer division*

```
sumPD(N)
  SUM=0
  loop I from 1 to N div 2
       //  Accept as upper boundary  N//2 OR N-1
       //   OR a java expression:(int)Math.round(Math.sqrt(N))
     if N mod I==0
       then SUM=SUM+I
     end if
  end loop
  return SUM
 end sumPD
```

***Note:** the subprogram heading and the return statement may not appear in the candidate's response.*

(c) *Award **[10]** max.*
*Award **[1]*** for *initializing indexes in all three arrays*
*Award **[1]*** for *correct outer loop*
*Award **[1]*** for *correct inner loop*
*Award **[1]*** for *calling the subprogram correctly*
*Award **[1]*** for *correctly accessing element of MAT (correct row and column indexes)*
*Award **[1]*** for *correct use of if statements*
*Award **[1]*** for *checking if MAT[R][C] is perfect and increasing the array PERFECT index*
*Award **[1]*** for *placing MAT[R][C] at correct place in array PERFECT*
*Award **[1]*** for *checking if MAT[R][C] is abundant and increasing the array ABUNDANT index*
*Award **[1]*** for *placing MAT[R][C] at correct place in the array ABUNDANT*
*Award **[1]*** for *checking if MAT[R][C] is deficient and increasing the array DEFICIENT index*
*Award **[1]*** for *placing MAT[R][C] at correct position in array DEFICIENT*

Example 1

```
IP=-1
ID=-1
IA=-1
loop R from 0 to 3    // Accept MAT.length() OR len(MAT) instead of 3
 loop C from 0 to 5// Accept MAT[R].length() OR len(MAT[R])instead of 5
           X = sumPD (MAT[R][C]) //calls the sumPD()subprogram
           if X == MAT[R][C]   //checking if perfect
           then
              IP=IP+1
              PERFECT[IP]= MAT[R][C]
           else if X > MAT[R][C]  //checking if abundant
                    then
                         IA=IA+1
                         ABUNDANT[IA]= MAT[R][C]
                    else
                         ID=ID+1
                         DEFICIENT[ID]= MAT[R][C]
                end if
            end if
 end loop
end loop
```

Example 2

```
IP=0
ID=0
IA=0
loop R from 0 to 3
  loop C from 0 to 5
      if sumPD(MAT[R][C])== MAT[R][C]
           then
               PERFECT[IP]= MAT[R][C]
               IP=IP+1
      end if
      if sumPD( MAT[R][C])> MAT[R][C]
            then
                ABUNDANT[IA]= MAT[R][C]
                IA=IA+1
      end if
      if sumPD( MAT[R][C])< MAT[R][C]
             then
                 DEFICIENT[ID]= MAT[R][C]
                 ID=ID+1
      end if
  end loop
end loop
```