8823 – 7011M

Markscheme

November 2023

Computer science

Higher level

Paper 1

18 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]****.*
Application software cannot run on a particular CPU architecture;
Software cannot run on an operating system;
Computer hardware components may not run with a CPU architecture/bus/motherboard;
Inability of software to interact with files;
Different data representation (units/conventions i.e., dates/currency, etc.);

Note: Reward other suitable answers.

2. *Award* ***[1 max]****.*
The NAND operator produces a FALSE value only if (and only if) both values of its two inputs are TRUE;

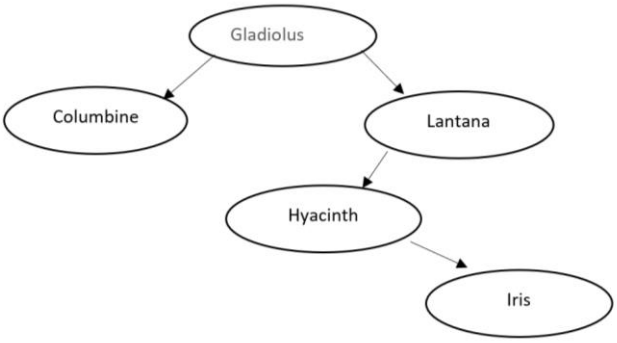
*Note**:** Accept a correct truth table for the expression A NAND B.*

3. *Award* ***[3 max]****.*
Fetches/extracts each instruction from memory;
Decodes/transforms them into several commands/signals/steps (that are passed to the ALU or I/O or other components in the CPU for execution);
Controls the movements of data within the CPU;
Generates the clock pulses that regulates speed of the instruction cycle;
Generates control signals for all hardware components to regulate their activities;
Synchronizes all the operations of the CPU;

Note: Accept other reasonable answers.

4. *Award* ***[3 max]****.*
Data;
Protocol;
Packet number;
Total number of packets;
Sender’s IP address;
Receiver’s IP address;
Control bits/Parity bit/Check digit/Time to live (hop limit);

5. *Award **[3 max]**.*
*Award **[1]** for the correct root*
*Award **[1]** for the correct left subtree*
*Award **[1]** for correct right subtree.*



*Note**:** Accept the mirror image.*

6. *Award **[2 max]**.*
A virtual machine (VM) is an operating system (OS) or application;
which imitates dedicated hardware (so that the end user has the same experience on a VM as they would on dedicated hardware);

A virtual machine (VM) is software that runs programs;
without being tied to a physical machine;

The virtual machine is a form of software that allows running an operating system (OS);
within another operating system;

Virtual machine is a software that allows running an OS other than the OS of the host machine;
while hiding/abstracting the physical environment/OS of the host machine;

***Note**: Accept answers related to the Java virtual machine.*

The VM is a platform independent execution environment;
that converts (Java) bytecode into machine language and executes it;

The (Java) virtual machine is a program that provides run-time environment;
in which (Java) byte code can be executed;

7. *Award **[2 max]**.*
number of rejected/hypothesized words (speech commands);
number of audio signal issues (users being either too loud, too slow or too fast/noisy environment – means decreased accuracy of speech recognition);
difficulty distinguishing between similar sounding words;
different dialects/accents which the system may not be able to deal with;
limited database of commands;

***Note**: Accept other reasonable answers.*

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

Concurrent processing means that a single or multiple system perform several tasks;
simultaneously/within overlapping time frames;

Concurrent processing means execution of several algorithms/programs/sub-programs;
At the same time (regardless of the number of processors);

(b) *Award **[1 max]**.*

Better use of (computer) resources;
Decreased response time/waiting time/increased efficiency;
Reduced overall run-time of the program;
More real-world problems can be solved;

9. (a) *Award **[4 max]**.*

fun(1216) = 6 + fun(121);
=6 + 1 + fun(12);
=6+1+2+ fun(1);
=6+1+2+1+ fun(0);
=6+1+2+1+0= 10;

***Note**: The working may be shown differently.*

(b) *Award **[2 max]**.*

Calculates/returns the sum;
Of all digits in `N`;

Section B

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

Passwords should be given to access certain aspects of the data;
There should be levels of hierarchy (for example, the receptionist will be only allowed to access data such as names, addresses but not medical history/doctor’s notes);

Multi-factor authentication/one-factor authentication;
which mandates for users (the doctor and receptionist) to verify their identities through various methods of validation (for example, PIN, incorporate thumb scanning or retina scanning technology)/which prevents random persons logging into the system;

Most security breaches are a consequence of human factors (negligence or simple “human” error);
This can be prevented by security awareness training for the doctor and receptionist/ precautionary measures when handling patient data;

Encryption;
makes it harder for hackers to decipher confidential patient data (if they manage to breach and subsequently gain access to the information);

Regularly installing updates/patches;
To ensure the data is protected against new threats;

(b) (i) *Award* ***[1 max]****.*

Human failure/error;
Software corruption;
Theft;
Computer viruses/other malware;
Hardware destruction;
Data corruption;

(ii) *Award* ***[2 max]****.*

Offsite/online backups/hard copy backups of patients' records should be made;
To be used in case of system failure;

With data loss prevention software/backup software;
That creates a file back/a copy doctor’s files to a secondary device in an automated fashion;

Firewall and antivirus software/all the software;
Need to be updated and properly maintained (to avoid corruption);

Training the staff with security practices;
to avoid human errors;

Physical security;
to protect against theft;

Maintaining the suitable conditions/following best practices for usage;
to avoid hardware destruction;

Note*: Reward other reasonable answers.*

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

A mail-merge feature can be used to contact them;
The receptionist types the body of the letter (template)/prepares the letter for the patients;
The database containing patient's records can be searched for the names and addresses of this type of patients;
To prepare the mailing list/table of recipients;

(d) *Award **[4 max]**.*
Security when working remotely;
VPN's data encryption features (putting data into a coded format so its meaning is obscured) allows the doctor to keep confidential information safe/VPN authenticates the user (doctor) before giving the access to data;

VPN allows access to any content in any place;
Because it enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network;

VPNs hide the location/hidden IP address;
making it seem as if she is accessing data from another place/location of the patient unknown for hackers/the MitM;

Protects the privacy of data;
A VPN will prevent apps and websites from attributing the doctor's behaviour to her computer's IP address/can limit the collection of the location and browser history from the Internet Service Provider(ISP);

VPNs usually have intuitive and user-friendly interface;
So, it makes installation and use easy for the doctor (non-technical user);

VPN is adaptable to many smart devices/protects smart devices such as phones, tablets and laptops;
So, the doctor can use various devices/any available device;

*Mark as **[2]** and **[2]***

(e) *Award **[3 max]**.*
the speed of data transmission across a mobile network can vary because of
the network technologies available in the area (3G, 4G, 5G);
because of limited bandwidth;
because the features of the doctor's device;
because of the amount of data being transmitted;

because the number of other users in the area using the same network (during certain hours, there are many users, which causes the connection speed to slow down);
because of the location of user with respect to cell towers (the speed may change because the signal varies depending on the coverage area);

11. (a) (i) *Award **[1 max]**.*
Surveys/questionnaire;
Observation;

(ii) *Award **[2 max]**.*
Note: The answer to a(ii) should align with the answer to a(i).

Surveys are a practical/fast way to gather information from a large group of people about something specific in the company (which gives a good composition of data)/ anonymity (confidentiality) thus allowing the respondents to express their opinions freely/surveys provide fast results because of today's mobile and online tools this method of data collection can generate results quickly**;**
it is difficult/time consuming/costly/ineffective to collect all the specific data interviewing all the end users/analysis of collected data will take more time**;**

Observation allows the entire graphic design company to account for all processes (parts/participants)/gives more realistic view than interview/can highlight aspects that are not detected in interviews/helps to speed up the work of collecting information/ helps to prove or disprove an idea while keeping the overall costs of the collection of information down;
compared to interviewing many end-users which is costly/time consuming, etc.;

Note: Reward other suitable answers.

(b) *Award **[2 max]**.*
Prototypes are simple representations of a system to demonstrate its functionality/user interface to the clients;
Used to evaluate a new design/to improve the accuracy/to collect data (with it the system analyst can survey clients/end-users to get feedback)/to ensure all essential functions of the system are present/meets the needs of the users;

(c) *Award **[3 max]**.*
Software needs to be tested with typical/extreme/invalid data so when the client (graphic design company) gets to use it most (or all) of the errors have been corrected;
this can save money (if bugs are caught in the earlier stages, it costs much less to fix them);
this adds to security/the client gets a trustworthy application (keeps client's information and data safe)/vulnerability free - problems and risks are eliminated beforehand;
this increases software quality/brings client satisfaction (best possible client experience);
ensures the application's compatibility on many devices and operating systems (companies usually have many devices and tools available);
helps the application come as intuitive and user friendly as possible;

(d) *Award **[3 max]**.*
False advertising;
Images manipulated/edited/falsified;
And used to deceive people into supporting a product/brand;

Forgery;
Images of jewellery/fashion products/buildings/landscapes could be edited/falsified;
For gain;

Fraud;
The image of an accident/happening can be changed/edited;
To mislead police/an insurance company/public;

Copyright violation/Copyright infringement/Copyfraud;
Copyrighted images used without permission of the copyright holder/without giving credit to the author;
to profit from their efforts/which is plagiarism/which is unlawful;

***Note**: Award marks for **any** identified and fully explained problem in areas such as advertising, journalism, fraud, pornography, offensive imaging, etc.*

(e) (i) *Award **[3 max]**.*
by compressing the file;
if the compressed file does not fit then split the file up into smaller parts;
then attach each of the individual pieces to separate emails;
the recipient would have to (download each attachment separately and then) use a file extraction program to extract the larger (divided-up) file, and piece it back together;

Accept examples, such as, use (file compression software like) 7-Zip to create an archive containing that 60MB file, splitting it into eight 7.5 MB pieces then use a file extraction program to extract file and piece it back together.

Use file-sharing services (Google Drive/OneDrive);
upload the large file to the cloud;
and share the link;

use an online storage service (that doesn't integrate directly with an email provider/Dropbox);
upload the large file and "share link";
then just paste the link into your email (and the recipient will be able to access the file by clicking it);

(ii) *Award **[1 max]**.*
Expensive hardware needed for high performance;
Need for special skills in use of graphic design software;
Unable to decode the file format;
File corruption;

***Note:** Accept hardware or software examples, such as expensive colour printing, expensive high-resolution screens, high storage requirements, complexity/cost/limitations of graphic design software.*

12. (a) (i) *Award **[2 max]**.*
Electromagnetic sensors;
Pressure sensors;
Infrared sensors;
Ultrasonic sensors;
Microwave sensors;
Inductive loop detectors;
Acoustic sensors;

(ii) *Award **[2 max]**.*
Sensors are appropriate because they automatically sense the presence in the given environment;
without the need for human operation/sensors can detect 24x7 without losing efficiency;

Sensors are appropriate because only presence of a car required;
this can be reliably detected by sensors/data is sent to CPU quickly enabling quicker reaction times;

Sensors are appropriate because it saves staffing the crossroads;
so, costs are reduced;

(b) *Award **[2 max]**.*
ROM;
Because it is non-volatile;

PROM;
can be programmed only once and is not erasable;

EPROM;
Can be reprogrammed in case of change in traffic flow;

EEPROM;
can be erased one byte at a time (rather than erasing the entire chip)/allows flexible process of reprogramming;

(c) (i) *Award **[2 max]**.*
An interrupt is a signal sent to the processor;
that halts the current process;

(ii) *Award **[3 max]**.*
The microprocessor responds to that interrupt with an interrupt handler/an ISR (Interrupt Service Routine)/a routine which instructs the microprocessor on how to handle the interrupt;
the routine can send a signal to output transducers;
to immediately turn the traffic lights to red to stop the cars;

The microprocessor receives an interrupt request and passes control to the interrupt handler;
it may wait a certain amount of time before sending the signal to output transducer;
to turn the traffic lights to red to stop the cars;

(d) (i) *Award **[2 max]**.*
Cameras improve traffic flow;
by controlling the movement of vehicles and determining traffic light timing;

Traffic cameras help to reduce accidents;
by controlling the movement of vehicles/identifying drivers who speed;

Cameras improve safety/have a preventive/disciplining effect;
can be used to identify defaulters and to take legal action/make them pay fine;

Installation and deployment of automated traffic cameras often pays for itself in the long run;
is still far less expensive than physical policing/traffic cameras can operate tirelessly 24/7;

Traffic cameras can be deployed with very little cost to infrastructure;
as they can run independently from the electrical grid only on solar power;

(ii) *Award **[2 max]**.*
Privacy is an issue;
as traffic cameras enable monitoring the activities of people (drivers, pedestrians);

It can be costly;
to install/maintain a large number of cameras and monitoring systems;

Cameras can be vulnerable to damage;
For example, wetness, moisture;

Cameras can be vulnerable to misuse;
for example, criminals/hackers might have worked out ways to disable or disconnect cameras;

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

A linked list is a data structure made of a chain of nodes, the external (head) pointer points to the first node and the link part of the last element of the list points to null/nil/none;
each node contains a data field (value) and a pointer (to the next node);

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

A queue data structure reflects the queuing which adds nodes at the end (enqueue) and deletes nodes from the beginning of the queue (dequeue)/FIFO data structure;

the data about the patient which is seen by a doctor would be deleted from the beginning of the queue;
and as patients arrive their data would be added to the end of the queue;
which is not really suitable because emergency cases may arise who need to be seen by a doctor immediately/first so the patients' data should be added at the beginning of the queue;
or a patient may decide to go home/leave the queue and the data should be deleted from the middle/end of the queue;

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

A new node for a new patient would be created and the data about the patient should be stored into the data field of the new node;
this new node should be inserted at the beginning of the singly linked list;
so, set the next pointer of the new node to point to the node pointed to by the external pointer/make the link part of the new node pointing to the existing first node;
then the external pointer to the list should be changed to make new node as the first node/ the external pointer should be updated to point to the new node;

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

Create a new node for a new patient and place the data about this patient into the data field of the node;
severity of illness defines **insertion position** of the node/where to place the node;
the new node should be inserted in the middle of the list (**after** the node containing data about the patient who has more serious symptoms and **before** the node containing data about the patient who has less serious symptoms than the new patient);
search the list for the node **prior** to the insertion position;
set the **next pointer of the new node** to point to the node pointed to by the next pointer of prior node;
set the **next pointer of the prior node** to point to the new node;

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

Size of a static data structure is fixed whilst dynamic structures are more flexible/size can be changed;
memory is allocated to a static data structure at compile time (is predetermined) whilst memory is allocated to the dynamic data structure as the program executes/at run-time;
elements of a static data structure can be directly accessed whilst access to the items in a dynamic data structure is sequential;
a dynamic structure come at extra costs of memory to store address pointers, so it uses more memory than a static structure (holding the same amount of data);
elements can be inserted into/deleted from a dynamic data structure just by manipulating pointers/by updating addresses without shuffling/shifting/resorting as in a static data structure;

14. (a) *Award **[1 max]**.*

*(*Judge #)3;

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

*Award **[1]** for a correct row loop*
*Award **[1]** for a correct column loop*
*Award **[1]** for correctly testing the array value*
*Award **[1]** for increasing COUNT*
*Award **[1]** for initializing and outputting COUNT*

```
COUNT=0
loop R from 0 to 19
      //accept MARKS.length -1
     // or len(MARKS)-1 instead of 19

    loop C from 0 to 7
          //accept MARKS[R].length-1
          // or len(MARKS[R])-1 instead of 7

          if MARKS[R][C]=10 then
               COUNT=COUNT + 1
          end if
    end loop
end loop
output ('The number of times the 10 score was awarded:', COUNT)
```

(c) *Award* ***[10 max]***

Example 1:
Award ***[1]*** *for the correct row loop (every competitor!)*
Award ***[1]*** *for initialization of SUM, HIGH and LOW within the outer loop and before the inner loop*
Award ***[1]*** *for the correct column loop (the competitor's marks!)*

Award ***[2 max]*** *for determining the highest value:*
Award ***[1]*** *for testing the array value with it*
Award ***[1]*** *for reassignment of HIGH*

Award ***[2 max]*** *for determining the lowest value:*
Award ***[1]*** *for testing the array value with it*
Award ***[1]*** *mark for reassignment of LOW*

Award ***[3 max]*** *for calculating total correctly:*
Award ***[1]*** *for adding array values to initialized variable (SUM)*
Award ***[1]*** *for subtracting (not adding) the highest value*
Award ***[1]*** *for subtracting (not adding) the lowest value*

Award ***[1]*** *for calculating average (total divided by 6)*
Award ***[1]*** *for outputting the competitor's name and the final score*

```
loop R from 0 to 19
    HIGH=MARKS[R][0]   //any value <= 0
    LOW= MARKS[R][0]   //any value >=10
    SUM=0
    loop C from 0 to 7
        SUM=SUM+ MARKS[R][C]
        if MARKS[R][C]>HIGH then
                HIGH= MARKS[R][C]
        else
           if MARKS[R][C]<LOW then
               LOW = MARKS[R][C]
           end if
        end if
    end loop
    SUM = SUM – HIGH - LOW
    FINALSCORE = SUM / 6
    output('Name:', NAMES[R])
    output('Final score:', FINALSCORE)
end loop
```

Note: *Three inner loops may be used to separately determine the highest score, the lowest score, and to calculate the sum.*

Example 2:
(A sorting algorithm used)

Award ***[1]*** *for correct row loop*
Award ***[1]*** *for creating/using temporary 1D array*

Award ***[5 max]*** *for sorting:*
Award ***[1]*** *for correctly using a flag*
Award ***[1]*** *for a correct outer loop*
Award ***[1]*** *for a correct inner loop*
Award ***[1]*** *for comparing DATA [I] with DATA [I+1]*
Award ***[1]*** *for the correct swap*
Award ***[1]*** *for correct use of array indexes*

Award ***[3 max]*** *for calculating total correctly:*
Award ***[1]*** *for adding array values to initialized variable (SUM)*
Award ***[1]*** *for subtracting (not adding) the highest value*
Award ***[1]*** *for subtracting (not adding) the lowest value*

Award ***[1]*** *for calculating average*
Award ***[1]*** *for outputting the competitor's name and the final score*

```
loop R from 0 to 19 //row loop
     //fills in a temporary array DATA with the row R values
     loop C from 0 to 7 //column loop
         DATA[C]=MARKS[R][C]
     end loop
     //arranges data values
     //from the lowest to the highest value
     //Bubble sort(can be any other sorting algorithm)
     F=True
     loop while F
          F=False
          loop I from 0 to 7-2
               if DATA[I] > DATA[I + 1] then
                    TEMP=DATA[I]
                    DATA[I] = DATA[I + 1]
                    DATA[I+1] = TEMP
                    F=True
               end if
          end loop
     end loop

     //HIGHEST is DATA[7]
     //LOWEST is DATA[0]
     //sums elements, except HIGHEST and LOWEST
     SUM = 0
     loop C from 1 to 6
           SUM=SUM + DATA[C]
     end loop

     FINALSCORE = SUM / 6
     output('Name:', NAMES[R])
     output('Final score:', FINALSCORE)
end loop //end row loop
```

***Note**: Bubble sort is on the syllabus. Other sorting algorithms are acceptable. In case any other sorting algorithm appears in a candidate’s response, contact your team leader for advice.*