N21/4/COMSC/HP1/ENG/TZ0/XX/M

# Markscheme

## November 2021

## 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.<br>• 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.<br>• 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]***

Primary storage is accessed by a computer's central processing unit (CPU) and secondary storage is not accessed directly by the CPU;
Primary storage has lower access time / has smaller capacity/ more expensive type of memory than secondary storage (which is slower than primary storage, with larger capacity but it is cheaper);
Primary storage is volatile (uses random-access memory (RAM), cache memory, or some other specialized hardware to store data while the computer is powered on/ volatile devices), whilst secondary storage on a computer is provided by non-volatile devices (such as SSD or HDD);
Primary storage holds data temporarily whilst data is kept permanently/ for a long time in secondary storage;
Primary storage holds currently running programs/data/operating system, secondary storage does not;

**2.** *Award* ***[1 max]***

8A;

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

A sorting algorithm is a method for reorganizing a number of items;
into a specific order (such as alphabetical order, highest-to-lowest value);

Sorting algorithm performs specific operations on the input list/array;
In order to deliver ordered list/array as output;

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

Bubble sort swaps adjacent items**;**
Selection sort finds the next smallest (each time it goes through the list);

Bubble sort can exit early/ is faster if already the list is sorted**;**
Selection sort will need to complete the procedure for the entire list every time;

(The efficiency of Bubble and selection sort is different when applied on **already sorted** list) in the best-case bubble sort takes an order of N time;
whereas selection sort consumes an order of $N^2$ time (where N is the number of items on the list));

***Note****: To award marks for such an answer it should be evident that the list is already sorted OR the term 'the best-case' should appear because the worst-case /average-case complexity/efficiency is same in both algorithms (O($N^2$)).*

**4.** *Award **[2 max]***
*Award **[1]** for showing the working out*
*Award **[1]** for the correct result (TRUE)*

((5>5) **XOR** TRUE) **AND** (3+2>4) = (FALSE **XOR** TRUE) **AND** TRUE=
=TRUE **AND** TRUE;
=TRUE;

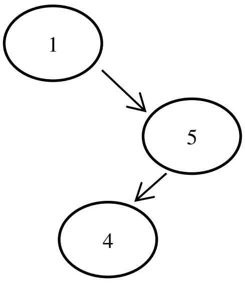
**5.** (a) *Award **[1 max]***

Binary tree (is a tree) in which every node has (no, one or) at most two children whilst a non binary tree can have nodes with more than 2 children (non binary trees do not have an upper limit on number of children nodes);

Each node in a binary tree can have at most two subtrees (left and right subtree), a node in a non binary tree can have any number of subtrees;

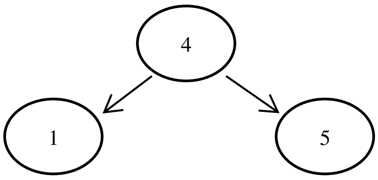
(b) *Award **[2 max]***

*Award **[1]** for the root (1)*
*Award **[1]** for the correct right subtree*



Alternative answer

*Award **[1]** for the root (4)*
*Award **[1]** for the correct right subtree*



**6.** *Award **[3 max]***

Compressing a data file reduces the file size;
The amount of time it takes to send a file over the network depends on the size of the transmitted file, so the amount of time needed to transmit the file is reduced /transmission of a compressed file takes less time than transmission of that same uncompressed file;
Also, compression of data files before transmission reduces the financial cost of running a network/ less equipment / bandwidth is needed to transmit the data file;

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

A sub-program is a named section of code that performs a specific task (in a program) / can be called by name / referred by the identifier when needed;
without knowing the details (of code and data structures) as these are wrapped / hidden within the sub-program;

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

different stages (of programming) (*Accept examples!*) run simultaneously (rather than consecutively);
this decreases product development time / decreases the time to market;
leading to improved productivity/reduces costs;
however, it requires more resources/more software developers;

(c) *Award **[2 max]***

A message can be sent to the user (When the software is installed and registered, the user provides an email address / phone number);
With a link to the update;

notifications/alerts are sent to the computer (a cookie is placed on the user's computer which communicates with the software developer);
about automatic updates;

(when the program is run) it queries a URL the program developer has built in
to check whether the current version matches the latest version;
if not, notifications/alerts are sent;

**8.** *Award **[3 max]***

OS keeps tracks of all peripheral devices (the I/O controller) / decides which process gets the device when and for how much time / allocates and de-allocates devices;
OS works with device drivers and the basic input/output system (BIOS) to perform hardware tasks / the necessary drivers (for every peripheral) are built into the OS and/or when a new peripheral is added software / device driver provided by a hardware maker is installed into the operating system (to tell the computer's OS how to work with the peripheral/hardware) (because without a device driver the OS would not be able to communicate with this peripheral device);
A device driver translates the OS's instructions into a language (analogue signals) that the device can understand;
There are various types of device drivers for peripherals (such as keyboards, mice, disk drives, controllers, printers, graphics cards, ports, etc.);
Device drivers run in the OS kernel space (in the part of the OS that directly interacts with the physical structure of the system) (and implement functions such as open, close, read, write);
The running application/user's program can make a call to device driver functions which provide an interface between user space and kernel space;

Example answer:
An OS manages peripherals via their respective drivers;
For example, sound card drivers;
are necessary so the OS knows exactly how to translate the 1s and 0s (that comprise the MP3 file) into audio signals;
that the sound card can output to headphones / speakers;

## Section B

**9.** (a) (i) *Award **[2 max]***

Surveys allow analysts to obtain appropriate information quickly;
from a large number of persons / stakeholders;

Standardized question formats are prepared;
that can provide data that can be easily quantified/allow quantitative analysis (the success of a survey depends on the effectiveness of questions);

The use of standardized formats;
minimizes the risk if the analyst adding their opinion;

Survey is a straightforward/easy/practical way of gathering data;
it can be distributed using various methods (printed copies, e-mail, embedded in a website, online forms)

Surveys allow greater anonymity for respondents;
which can lead to more honest responses;

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

Poorly designed surveys;
May make it difficult to analyse the data;
(Response may be limited/not all questions answered / not all forms will be returned;
Can be challenging to analyse the collected data ;)

Questions in the survey were not effective;
Can be differences in how people understand the survey questions;
Data gathered cannot be quantified;

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

Direct observation;
Interviewing;
Focus groups;
Examining existing documents/ Literature searches;
Investigating previous solutions;

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

The final product meets the requirements/ is more successful;
as feedback is provided by the users during the development process;

Costs/time saved at a later stage;
as early feedback avoids later changes (which may require a considerable amount of time/cost);

The best design can be decided upon early;
As it allows different prototypes can be tried out/tested;

*Mark as 2 and 2.*

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

Another cycle of analysis and design might be needed because the stakeholder could ask for modifications;
because errors or omissions are found that need to be corrected;
new/different features could be added that affect the current design;
which affects the costs/ delivery time / hardware requirements/ contributes significantly to system quality and performance;

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

so that it functions the way it is supposed to / to ensure that the actual outcomes are equal to the predicted outcomes;
so that it meets its design specifications functions;
so that functions correctly / to eliminate any errors/bugs;
so that the speed/capacity/compatibility issues are solved;
so that all security features are configured and enabled;
because wrong/incorrect/stolen information could have serious consequences for its customers (accept an actual consequence);
because wrong information could harm the company's reputation / loss of earnings / sued/ etc;

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

Each user should have appropriate access rights;
Managers, employees and clients/customers are permitted to access different parts of the data;

Sensitive data can be protected by locking it with a password;
preventing unauthorised access who doesn't have this;

Administrator should record who has logged on and from which computer / and for how long;
to discourage security violations/ to avoid undesirable events from occurring;

File systems should be encrypted (as it passes throughout a network / resides on computers);
to make data unusable if accessed by unauthorized user;

Each computer should have access rights depending on its location;
Logged in computers should not be unattended;

*Mark as 2 and 2.*

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

Each user should have a user ID and a personal password;
Passwords should be regularly changed;

Two-factor authentication for remote users and administrators could be required (this could be a digital certificate, tokens, thumbprint scanners);
In addition to the usual user ID and password;

In order to prevent intruders/strangers from accessing the company's network;
the router is set to accept only specific MAC addresses;

Regularly installing updates and patches;
to ensure the network is protected against new threats such as malware;

*Mark as 2 and 2.*

(c) *Award* ***[4 max]***
Restore files from a backed-up data file;
ensuring it is a recent copy to minimize loss;
a parallel/failover system could be operated;
that could be switched to (if the live system is corrupt);
data recovery software could be (installed and) run;
to repair corrupted files;

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

A VPN improves data security;
By passing the company’s data through a hidden tunnel;
And encrypting the internet traffic inside encapsulated data packets;

A VPN improves the company’s productivity;
as the workers will not have to be in a particular location to get to be productive;
VPN is not dependent on any particular network or Wi-Fi connection to work/ can be used on any type of device;

Remote access;
VPNs is not dependent on any particular network or Wi-Fi connection to work;
The company can have a remote workforce /employees or freelance staff working from different geographic locations/can connect their different office locations;

Some databases/websites that support the company’s business operations may not be directly accessible in some countries;
A VPN helps to unblock geo restricted contents / has ability to bypass geo-blocking;
By hiding IP addresses / obscuring the access requests to appear to be originating from various IP address which are not in an unrestricted location;

A VPN can reduce a company’s infrastructure costs;
Because can be used on any type of device;
So, the company can offer BYOD options to employees;

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

The GPS receiver in a train takes the information from the satellite/ picks up the signals from (at least) 3 satellites;
The signals transmitted are: time of transmission, coordinates of the satellite;
The difference between the (atomic) time of transmission and (atomic) time of receiving the signal is used to calculate the distance;
The method of trilateration is used to determine position from the distance to satellites/ is used to determine the train's exact position/ is used to calculate position of the train through equation resolution on a sphere;

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

Train number/ route / direction;
Status (on time / delayed / cancelled);
Time train / **next** train are due;

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

Motion sensors/vibration sensors (can be placed on track rail);
(*Accept any sensors which detect trains that pass them!)*

(Sensing devices are placed) at a distance from each side of a train station/junction
OR
at some fixed points where obstacles (like tunnels) prevent GPS from working properly;

Data from each sensor can be logged continuously and used in calculations/ determinations (of expected arrival times/ delays / rerouting during disruption, etc.);

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

Real-time information (train number/ route / direction and status (on time / delayed / cancelled);
relating to the arrival / departure of a particular train;

The time next train is due (the next 2/3 times a train will arrive at the station);
helps passenger to spend less time waiting on trains;

Should save user's (passenger) most frequented stops and routes;
for quick access to get the next rail trip;

Alerts (set);
to let you know when a train is nearing the station;

Trip planning (conducted);
through Google trip planner within the app;

Information on the nearest station, distance to station/ next train arrival/ next train departure;
to help passenger to spend less time travelling/ save time;

*Mark as 2 and 2.*

(d) *Award **[4 max]***
Safer transit;
Trains are tracked and in case of emergency the location can be reached at the earliest and emergency services provided;

More favourable view of transit/helps passenger to get punctuality status of the train/;
GPS can provide worldwide three-dimensional positions (24 hours a day) (in any type of weather);

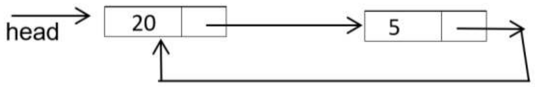
Sometimes it is too difficult to ensure reliable positioning;
Objects, such as buildings, overpasses, and other obstructions (that shield the antenna from a satellite) can potentially weaken a satellite's signal;

Privacy issues;
Knowing the absolute position of anything, anytime, anywhere;

*Mark as 2 and 2.*

**12.** (a) *Award **[3 max]***

*Award **[1]** for the node containing number 20 is pointed to by the pointer 'head';*
*Award **[1]** for the diagram showing only two nodes and all the correct links;*
*Award **[1]** for the last node containing number 5;*



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

The last node is identified by its next/link pointer;
which contains the address of the node at the beginning of the list / is equal to the pointer “head “;

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

Initialize (a temporary pointer with the head and) a variable sum with 0 (zero);
Loop from the beginning to the end of the circular linked list / until all the nodes get traversed);
Add the value (of the data field) (of the current node) to the sum;
Change the temporary pointer so it points the next node of circular linked list;

***Note:*** *Answers written in pseudocode are acceptable.*
*For example,*

```
sum = 0
curr = head
loop
  sum = sum + curr.data
  curr = curr.next
until curr == head
```

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

In an array, memory is assigned during compile time(predetermined) whilst in a linked list it is allocated during execution/runtime;
Arrays are of fixed size whilst linked lists are flexible and can expand and contract its size;
Array requires less memory (due to actual data being stored within the index in the array) whilst there is a need for more memory in linked list (due to storage of additional next and previous pointers/references);
Elements are stored consecutively in array whereas elements are stored randomly in linked lists;
Memory utilization is inefficient in the array whilst memory utilization is efficient in the linked list;
Accessing an element in an array is direct (fast) while accessing an element in linked list is sequential/linear (slower) /to get into the nth element only the array name with index within the square bracket should be written whilst a linked list (to get the nth element) should be traversed starting from the beginning of the list, and traversing through the list until found/ ;
Operations like insertion and deletion in arrays consume a lot of time whilst the performance of these operations in linked lists is fast;

(e) *Award **[2 max]***

Print queues (a number of print jobs on a queue instead of waiting for each one to finish before specifying the next one);
Queue is used for synchronization when data is transferred asynchronously between two processes (for example IO Buffers, file IO);
Queues are used in CPU scheduling algorithms;
Handling of interrupts in real-time systems (the interrupts are handled in the same order as they arrive, first come first served);
Computer modelling of physical queues (supermarket checkouts) (call centre phone systems use queues to hold people calling them in an order, until a service representative is free);

**13.** (a) *Award **[1 max]***
5.9;

(b) *Award **[5 max]***
*Award **[1]** for correct outer/row loop*
*Award **[1]** for correct inner/column loop*
*Award **[1]** for use of a flag*
*Award **[1]** for checking whether all elements on and above the main diagonal are zero*
*Award **[1]** for checking all elements below the main diagonal (they all should be positive numbers)*
*Award **[1]** for outputting the appropriate message*

Example 1:

```
VALID=True
loop R from 0 to 9
   loop C from 0 to 9
        if R>C and ROUTE_X_DISTANCES[R][C]<=0
             then VALID=False
         end if
         if R<=C and ROUTE_X_DISTANCES[R][C]!=0
             then VALID=False
         end if
     end loop
end loop
if VALID
  then output('VALID')
  else output('INVALID')
end if
```

Example 2:

```
FLAG=1
loop R from 1 to 9
     loop C from 0 to R-1
         if ROUTE_X_DISTANCES[R][C]<=0
             then FLAG=0
         end if
     end loop
end loop

loop R from 0 to 9
     loop C= from R to 9
         if ROUTE_X_DISTANCES[R][C]!=0
             then FLAG=0
         end if
     end loop
end loop
if FLAG ==1
  then output('IT IS VALID')
  else output('IT IS NOT VALID')
end if
```

Example 3:

***Note**: Marks should also be awarded if a candidate wrote the algorithm in Java/Python/Javascript.*

*Award [1] for correct outer/row loop*
*Award [1] for correct inner/column loop*
*Award **[1]** for stopping as soon as an incorrect value is found*
*Award **[1]** for checking whether elements on and above the main diagonal are zero*
*Award **[1]** for checking elements below the main diagonal (they all should be positive numbers)*
*Award **[1]** for outputting the appropriate message*

```
function check()

{ for (var i=0; i<10; i++)
  { for (var j=0; j<10; j++)
    { if (i>j)
      { if (ROUTE_X_DISTANCES[i][j] <= 0.0) return "invalid";
      }
      else if (ROUTE_X_DISTANCES[i][j] != 0.0 ) return "invalid";
    }
  }
  return "valid";
}


output("ROUTE_X_DISTANCES is "+check());
```

(c*)* *Award* ***[6 max]***
*Award* ***[1]*** *for all variables correctly declared and initialized;*
*Award* ***[1]*** *for looping through the array ROUTE_X_NAMES;*
*Award* ***[1]*** *for determining positions of the first name in the array;*
*Award* ***[1]*** *for determining positions of the second name in the array;*
*Award* ***[1]*** *for outputting a message if one or other not present;*
*Award* ***[1]*** *for a comparison of positions to find largest;*
*Award* ***[1]*** *for the correct output of distance from ROUTE_X_DISTANCES;*

Example 1:

```
NAME1=input()
NAME2=input()
POS1=-1
POS2=-1
K=0
loop while  K<=9 and (POS1==-1 or POS2==-1)
      if ROUTE_X_NAMES [K].equals(NAME1) //accept '==' instead of equals()
        then POS1=K
      end if
      if ROUTE_X_NAMES [K].equals(NAME2)
         then POS2=K
      end if
      K=K+1
end while
if POS1==-1 OR POS2==-1
   then output('stations are not found')
   else
      if POS1 > POS2
         then output(ROUTE_X_DISTANCES [POS1][POS2])
         else output(ROUTE_X_DISTANCES [POS2][POS1])
      end if
end if
```

Example 2:

```
ST1=input()
ST2=input()
PS1=-1
PS2=-1
loop K from 0 to 9
      if ROUTE_X_NAMES [K]==ST1
        then PS1=K
      end if
      if ROUTE_X_NAMES [K]==ST2
         then PS2=K
      end if
end loop
if PS1!=-1 AND PS2!=-1
  then   if PS1 < PS2
         then T=PS1
              PS1=PS2
              PS2=T
         end if
         output(ROUTE_X_DISTANCES [PS1][PS2])
   else
         output('stations not found')
end if
```

Example 3:

***Note**: Award marks if algorithm is presented in a Java/Python/Javascript/any other program rather than IB pseudocode.*
*For example, please see the following Javascript program*

```
function findStation(station)
{ var found = false;
  var i = 0;
  do
  { found = (ROUTE_X_NAMES[i] == station);
    if (!found) i = i + 1;
  } while (!found && i < 10);
  if (found) return i;
  else
  { output("No such bus station as "+station);
    return -1;
  }
}



var station1 = input();
var station2 = input();
output("Finding the distance between "+station1+" and "+station2);
var station1index = findStation(station1);
var station2index = findStation(station2);
if (station1index >=0 && station2index >= 0)
   { if (station1index >= station2index)
       output("Distance "+ROUTE_X_DISTANCES[station1index][station2index]);
   else
    output ("Distance = "+ROUTE_X_DISTANCES[station2index][station1index]);
   }
```

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

Determine positions/indexes/subscripts of both bus stations in array ROUTE_X_NAMES;

Calculate the sum of the elements of array ROUTE_X_TIMES (calculate the number of minutes as the sum of the array elements);

Between (lower +1) index and higher index;