

# Markscheme

November 2021

Computer science

Standard level

Paper 1

13 pages

© International Baccalaureate Organization 2021

All rights reserved. No part of this product may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without the prior written permission from the IB. Additionally, the license tied with this product prohibits use of any selected files or extracts from this product. Use by third parties, including but not limited to publishers, private teachers, tutoring or study services, preparatory schools, vendors operating curriculum mapping services or teacher resource digital platforms and app developers, whether fee-covered or not, is prohibited and is a criminal offense.

More information on how to request written permission in the form of a license can be obtained from <https://ibo.org/become-an-ib-school/ib-publishing/licensing/applying-for-a-license/>.

© Organisation du Baccalauréat International 2021

Tous droits réservés. Aucune partie de ce produit ne peut être reproduite sous quelque forme ni par quelque moyen que ce soit, électronique ou mécanique, y compris des systèmes de stockage et de récupération d'informations, sans l'autorisation écrite préalable de l'IB. De plus, la licence associée à ce produit interdit toute utilisation de tout fichier ou extrait sélectionné dans ce produit. L'utilisation par des tiers, y compris, sans toutefois s'y limiter, des éditeurs, des professeurs particuliers, des services de tutorat ou d'aide aux études, des établissements de préparation à l'enseignement supérieur, des fournisseurs de services de planification des programmes d'études, des gestionnaires de plateformes pédagogiques en ligne, et des développeurs d'applications, moyennant paiement ou non, est interdite et constitue une infraction pénale.

Pour plus d'informations sur la procédure à suivre pour obtenir une autorisation écrite sous la forme d'une licence, rendez-vous à l'adresse <https://ibo.org/become-an-ib-school/ib-publishing/licensing/applying-for-a-license/>.

© Organización del Bachillerato Internacional, 2021

Todos los derechos reservados. No se podrá reproducir ninguna parte de este producto de ninguna forma ni por ningún medio electrónico o mecánico, incluidos los sistemas de almacenamiento y recuperación de información, sin la previa autorización por escrito del IB. Además, la licencia vinculada a este producto prohíbe el uso de todo archivo o fragmento seleccionado de este producto. El uso por parte de terceros —lo que incluye, a título enunciativo, editoriales, profesores particulares, servicios de apoyo académico o ayuda para el estudio, colegios preparatorios, desarrolladores de aplicaciones y entidades que presten servicios de planificación curricular u ofrezcan recursos para docentes mediante plataformas digitales—, ya sea incluido en tasas o no, está prohibido y constituye un delito.

En este enlace encontrará más información sobre cómo solicitar una autorización por escrito en forma de licencia: <https://ibo.org/become-an-ib-school/ib-publishing/licensing/applying-for-a-license/>.

**Subject details: Computer science SL 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 45 marks.

Maximum total = 70 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	<ul style="list-style-type: none"> <li>• In the case of an “identify” question, read all answers and mark positively up to the maximum marks. Disregard incorrect answers.</li> <li>• In the case of a “describe” question, which asks for a certain number of facts <i>eg</i> “describe two kinds”, mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications.</li> <li>• In the case of an “explain” question, which asks for a specified number of explanations <i>eg</i> “explain two reasons ...”, mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation <i>etc.</i></li> </ul>

## Section A

1. *Award [2 max]*

Screen size/ larger screen;  
Easier to view large amounts of data without excessive scrolling / reduced eyestrain / more accessible to those with weak eyesight;

Screen size/smaller screen;  
Difficult to accurately select the correct character on the virtual keyboard;

2. *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.

3. *Award [2 max]*

because the binary system uses only two digits (0 and 1);  
and computer uses bi-stable devices / devices that can be either in one of the two possible states (such as pulse or no pulse, on or off, 0 or 1, true or false) /;

4. *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;

5. (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;

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

The selection sort algorithm maintains two subarrays of a given array, the subarray which holds already sorted elements;  
And the subarray which is unsorted;  
In every iteration, the minimum element (considering ascending order OR maximum element considering descending order) from the unsorted subarray is found/selected;  
And then swapped with the item in the next position (at the appropriate place / in the last position) to be filled in the sorted subarray;

**OR**

The selection sort algorithm searches through the entire array for the smallest (considering ascending order OR the largest element considering descending order) element;  
When found, it (the smallest/largest element) is swapped with the first element of the array;  
Then searches for the smallest / largest element in the **remaining** array (an array without the first element) and swap it with the second element;  
This process **repeats** on the remaining items (searches for the smallest / largest element in the remaining array (an array without first and second elements)) and swap it with the third element, and so on);

(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<sup>2</sup> 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<sup>2</sup>)).

7. **Award [1 max]**

(5<=5) XOR (3>5) (= TRUE XOR FALSE ) = TRUE;

8. **Award [3 max]**

**Award [1]** for correct values in column I

**Award [1]** for correct values in column K

**Award [1]** for correct output

**Note:** Column 6 > I may not appear in the trace table.

K	I	6 > I	output
0	0	true	
0	2	true	
2	4	true	
6	6	false	6

## Section B

### 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]**

**Award [1]** mark for correct condition (in if statement) and correct calculation of the cost per km for each type of tickets, **x4**.

Example 1:

```

costperkm(AGE)
  if AGE<5
    then COST=0.0
  else
    if AGE<=15
      then COST=0.20*0.50          // COST=0.20*50/100
    else
      if AGE<65                    //accept AGE<=65
        then COST=0.20
      else COST=0.20*0.70          // COST=0.20*70/100
      endif
    endif
  endif
  return COST
end costperkm

```

Example 2:

```

costperkm(AGE)
  if AGE<5
    then COST=0.0
  endif
  if AGE>=5 and AGE<=15
    then COST=0.20*0.50          // COST=0.20*50/100
  endif
  if AGE>=15 and AGE<65
    //accept AGE>=15 and AGE<=65
    then COST=0.20
  endif
  if AGE>=65
    //if condition in the previous if statement is
    //AGE>=15 and AGE<=65, then this condition
    //should be AGE>65
    then COST=0.20*0.70          // COST=0.20*(100-30)/100
  endif
  return COST
end costperkm

```

**Note:** The method heading and the return statement may not appear in the candidates' responses.

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

1.3;

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

1.2 + 1.0 or 2.2;

(d) **Award [3]**

The sum of elements;  
in the array DISTANCES from P1+1;  
to P2;

(e) **Award [6 max]**

**Award [1]** for all variables correctly declared and initialized;

**Award [1]** for any correct loop through the array NAMES;

**Award [1]** for to determine positions of **both** names in the array;

**Award [1]** for outputting a message if one **or** other not present;

**Award [1]** for comparison of positions to find smallest/largest;

**Award [1]** for correct calculation and output of the price;

**Award [1]** for correct invocation of methods `calcdistance()` and `costperkm()`

**Example 1:**

```

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

```

**Example 2:**

```

ST1=input()
ST2=input()
AGE= input()
PS1=-1
PS2=-1
loop K from 0 to 9
    if NAMES[K].equals(ST1) //accept '==' instead of equals()
        then PS1=K
    end if
    if NAMES [K].equals(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
        PRICE= calcdistance(POS1,POS2) * costperkm(AGE)
        output(PRICE)
    else
        output('stations not found')
    end if
end if

```

**Example 3:**

**Note:** Award marks for the algorithm written in Java/Python/Javascript/any other programming language. The following example is the solution written in Javascript.

```

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 P1 = findStation(station1);
var station2 = input();
var P2 = findStation(station2);
var AGE = input();
if (P1 >= 0 && P2 >=0)
{ if (P1 < P2)
  output("Cost = "+costperkm(AGE)*calcdist(P1, P2));
else
  output("Cost = "+costperkm(AGE)*calcdist(P2, P1));
}

```