Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

May 2021

Computer science

Standard level

Paper 1

14 pages

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	• 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]***
Client;
Server/email server/DNS server/file server;
Router;
Firewall; **[2]**

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

Parallel;
old system and new system are operated at the same time until the current system is proved to be successful;

Pilot;
the new (whole) system is operated in one branch/part of the organization before it is rolled out to the whole organization;

Direct;
the new system replaces the old system in an immediate switchover;

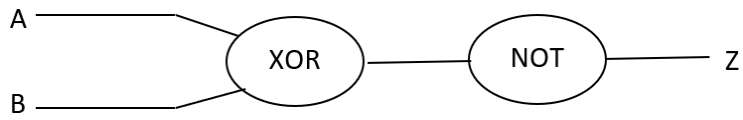
Phased;
the new system in phases / stages, gradually replaces parts of the old system until the current system is completely replaced by the new system;

[2]

3. *Award **[2 max]***

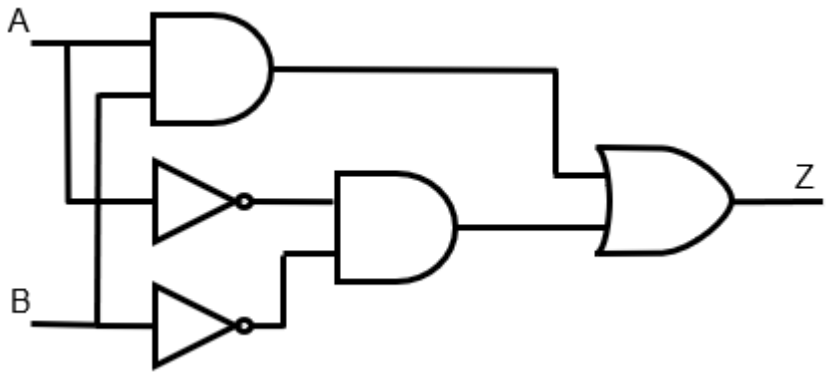
***Note:** There could be many answers that are correct.*

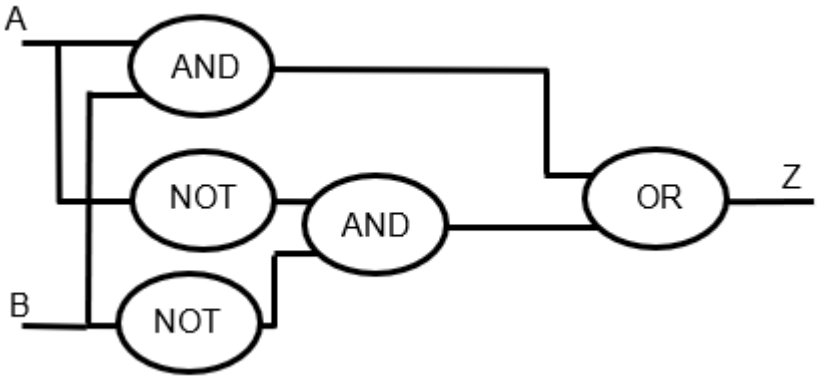
Example 1



Correct inputs and XOR gate;
NOT gate, correct final output and link from XOR gate; **[2]**

Example 2





Award [2] max.

2 marks, 1 mark for each correct input in OR gate
Note:** in this example the two inputs in OR gate are (NOT A AND NOT B)and (A AND B**).
***1** mark for drawing any 3 gates (complete with inputs and outputs)*

4. **(a)** *Award **[2 max]***
Allows bugs/error in operating system to be repaired;
Allows new features to be added to operating system (such as security updates, improving functionality, improving usability, etc.);
Allows compatibility issues to be improved; **[2]**

(b) *Award **[2 max]***
Automatic patches/updates sent (via internet);
User requested updates (via internet);
Patches sent on CD/DVD/memory stick; **[2]**

5. *Award **[2 max]***
11 x 16 + 15;
191;

Allow solution via binary route 1 mark for working, 1 mark for answer.
Allow both marks if correct answer given. **[2]**

6. *Award **[2 max]***
Fibre optics allow faster transmission speeds;
Fibre optic cables are more secure/harder to break into;
Fibre optic cable transmission is more reliable/less likely to suffer interference;
Fibre optics allow transmission over longer distance;
Fibre optics allow greater bandwidth;

[2]

7. *Award **[2 max]***
The value of a variable can change while value of a constant does not change;
During program execution/ during run-time / while stored in the memory;

[2]

8. *Award **[3 max]***
6;
12;
18;

[3]

9. *Award **[1 max]***
A unit of data made into a single unit (that travels along a given network path);
A data packet travels through a network as a unit *ie* with all parts kept together;

[1]

10. *Award **[1 max]***
Toolbars;
Menus;
Dialogue boxes; **[1]**

11. *Award **[4 max]***

1 mark: 1–2 correct rows;
2 marks: 3–4 correct rows;
3 marks: 5–6 correct rows;
4 marks: 7–8 correct rows;

A	B	C	P (A')	Q (P.B)	R (B+C)	Z (R.Q)
0	0	0	1	0	0	0
0	0	1	1	0	1	0
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	1	0
1	1	1	0	0	1	0

[4]

Section B

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

(Wireless) router;
A central hub for all the computers to connect to;
Enables wireless network packet forwarding and routing;

Wireless Network Interface Card (NIC);
To allow the computer to 'talk to' the (wireless) router;

Wireless access points;
allow Wi-Fi devices to connect to a wired network;

Wireless repeaters;
To expand the reach of the network;

Mark as 2 and 2. **[4]**

(b) *Award **[2 max]***
The ability to use their own devices at school;
The ability to access the school network from anywhere in the school;
No cables laid, so reduces the risk of tripping over cables;
Numbers of connections are not limited to cable ports, so greater numbers of students can connect at any given time;

[2]

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

Use of encryption;
So that data cannot be understood if it is intercepted;

Use of user authentication/usernames and passwords;
To prevent unauthorized access to the system;

Setting up a file of accepted MAC addresses;
To only allow access to the network by registered mobile devices;

Hide network ID;
So that the wireless network is not publicly seen;

Mark as 2 and 2. **[4]**

(d) *Award **[2 max]***
Client VPN software (to make a secure remote connection);
VPN-aware routers and firewalls (to permit VPN traffic to pass);
VPN appliance/server (to handle incoming VPN traffic);
Encryption protocol IPSec or SSL; **[2]**

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

Enhanced **s**ecurity of data;
for example, using encryption;
This prevents unauthorised access;

Remote access to data and resources (from any location);
Normal access of materials on the network;
as though the user was using the network on site; **[3]**

13. (a) *Award **[4 max]***
Collection method `NAMES.getNext()` / `NAMES.getData()` correctly used;
Correct loop;
Correct use of index (in both arrays);
Correct assignment in array for surnames;
Correct assignment in array for first names;

Note:
Award 1 mark in case that string methods are used to separate 'name' and 'surname' in the data item.

Example answer 1:

```
NAMES.resetNext() // reset and
SURNAME[600]      //initialization of arrays
FIRSTNAME[600]   //may not appear in candidates' responses
COUNTER = 0
loop while NAMES.hasNext()
    SURNAME[COUNTER] = NAMES.getNext()
    FIRSTNAME[COUNTER] = NAMES.getNext()
COUNTER = COUNTER + 1
end loop
```

Example answer 2

```
NAMES.resetNext()
loop COUNTER from 0 to 599
    SURNAME[COUNTER] = NAMES.getNext()
    FIRSTNAME[COUNTER] = NAMES.getNext()
end loop
```

Example answer 3 (assumes that items in the collection are objects- two attributes: surname and firstname)

```
I = 0
loop while NAMES.hasNext()
        X= NAMES.getData()
        SURNAME[I] = X.surname
        FIRSTNAME[I] = X.firstname
        I = I + 1
end loop
```

[4]

(b) *Award **[5 max]***
Correct outer loop;
Correct inner loop;
Checking of surname order;
Swapping surnames if necessary;
Swapping corresponding names;
Correct use of flag;

Example 1:

```
loop I from 0 to 599
   loop C from 0 to 598-I  //accept 598
       if SURNAME[C] > SURNAME[C + 1] then
         TEMP1 = SURNAME[C]
         TEMP2 = FIRSTNAME[C]
         SURNAME[C] = SURNAME[C + 1]
         FIRSTNAME[C] = FIRSTNAME[C + 1]
         SURNAME[C + 1] = TEMP1
         FIRSTNAME[C + 1] = TEMP2
       end if
    end loop
end loop
```

Example 2:

```
FLAG = TRUE
loop while FLAG = TRUE
    FLAG = FALSE
    loop COUNTER from 0 to 598
       if SURNAME[COUNTER] > SURNAME[COUNTER + 1] then
         TEMP1 = SURNAME[COUNTER]
         TEMP2 = FIRSTNAME[COUNTER]
         SURNAME[COUNTER] = SURNAME[COUNTER + 1]
         FIRSTNAME[COUNTER] = FIRSTNAME[COUNTER + 1]
         SURNAME[COUNTER + 1] = TEMP1
         FIRSTNAME[COUNTER + 1] = TEMP2
         FLAG = TRUE
       end if
    end loop
end loop
```

[5]

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

Example 1:

Calculate the index of the middle point in the array SURNAME: (first + last)/2;
Compare surname found with the one stored at middle point;
If greater than the value at the middle point, search the upper half of the array (right side) by calling the binary search method again with a new first index (first = middle + 1);
If smaller than the value at the middle point, search the lower half of the array (left side) by calling the binary search method with a new last (last = middle -1);
if found algorithm terminates;

Example 2:

Find the centre point of the array SURNAME[ ];
Compare surname to be found with the current name in SURNAME[ ];
If correct surname found => STOP;
Else if surname to be found is greater than the current name in SURNAME[ ]
eliminate lower half of array from search and repeat algorithm;
Else if surname to be found is less than the name in SURNAME[ ] eliminate upper half of array from search and repeat algorithm;

Note: Allow a mark for provision for name not found;

[4]

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

1 mark for a benefit (such as reusability, modularity, maintainability, readability)
1 mark for an expansion.

Example answer:
These sub-programs (sorting/searching) could be used in (many) other programs;
Which saves programmer's time/ effort;

[2]

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

NUMBER	DIGIT	OUTPUT
19	1	1
9	1	1
4	0	0
2	0	0
1		1

*Award up to **[4 max]**:*
Correct table with correct first row;
Rest of NUMBER column correct;
Rest of DIGIT column correct;
Rest of OUTPUT column correct; **[4]**

(b) Award ***[1 max]***
Converts denary number to (reverse) binary. **[1]**

(c) *Award **[6 max]**:*
Validation to check for positive numbers;
Validation to check for integers;
Loop for validation rules;
Loop structure to continue while NUMBER > 1;
Correct calculations/outputs inside loop;
Correct output of last digit outside loop;

Example answer:

```
input NUMBER
loop while NUMBER <= 0 OR NUMBER ≠ NUMBER div 1
    output "Please input another number, your entry is invalid "
    input number
end loop
loop while NUMBER > 1
    DIGIT = NUMBER mod 2
    output DIGIT
    NUMBER = NUMBER div 2
end loop
output NUMBER
```

[6]

(d) *Award **[4 max]***
The algorithm could use loops;
To remove the necessity to process extra lines of code;

Use of arrays/data structures;
So data can be stored/re-used/re-entered;

Use of flags;
To stop a search routine when an item has been found so that all elements don't have to be searched;

Mark as 2 and 2 **[4]**