



# **MARKSCHEME**

**May 2008**

**COMPUTER SCIENCE**

**Higher Level**

**Paper 2**

*This markscheme is **confidential** and for the exclusive use of examiners in this examination session.*

*It is the property of the International Baccalaureate and must **not** be reproduced or distributed to any other person without the authorization of IB Cardiff.*

## General Marking Instructions

*After marking a sufficient number of scripts to become familiar with the markscheme and candidates' responses to all or the majority of questions, Assistant Examiners (AEs) will be contacted by their Team Leader (TL). The purpose of this contact is to discuss the standard of marking, the interpretation of the markscheme and any difficulties with particular questions. It may be necessary to review your initial marking after contacting your TL. **DO NOT BEGIN THE FINAL MARKING OF YOUR SCRIPTS IN RED INK UNTIL YOU RECEIVE NOTIFICATION THAT THE MARKSCHEME IS FINALIZED.** You will be informed by e-mail, fax or post of modifications to the markscheme and should receive these about one week after the date of the examination. If you have not received them within 10 days you should contact your TL and IB Cardiff. Make an allowance for any difference in time zone before calling. **AEs WHO DO NOT COMPLY WITH THESE INSTRUCTIONS MAY NOT BE INVITED TO MARK IN FUTURE SESSIONS.***

You should contact the TL whose name appears on your “Allocation of Schools listing” sheet.

### **Note:**

Please use a personal courier service when sending sample materials to TLs unless postal services can be guaranteed. Record the costs on your examiner claim form.

## General Marking Instructions

1. Once markscheme is received mark in pencil until final markscheme is received.
2. Follow the markscheme provided, do **not** use decimals or fractions and mark only in **RED**.
3. Where a mark is awarded, a tick (✓) should be placed in the text at the **precise point** where it becomes clear that the candidate deserves the mark.
4. Sometimes, careful consideration is required to decide whether or not to award a mark. Indeed, another examiner may have arrived at the opposite decision. In these cases write a brief annotation in the **left hand margin** to explain your decision. You are encouraged to write comments where it helps clarity, especially for moderation and re-marking.
5. Unexplained symbols or personal codes/notations on their own are unacceptable.
6. Record subtotals (where applicable) in the right-hand margin against the part of the answer to which they refer. Show a mark for each part question (a), (b), *etc.* Do **not** circle sub-totals. Circle the total mark for the question in the right-hand margin opposite the last line of the answer.
7. Where an answer to a part question is worth no marks, put a zero in the right-hand margin.
8. Record the mark awarded for each of the four questions answered in the Examiner Column on the cover sheet.  
Add up the marks awarded and enter this in the box marked TOTAL in the Examiner Column on the cover sheet.
9. After entering the marks on the cover sheet check your addition of all marks to ensure that you have not made an arithmetical error. Check also that you have transferred the marks correctly to the cover sheet. **We have script checking and a note of all clerical errors may be given in feedback to all examiners.**
10. Every page and every question must have an indication that you have marked it. Do this by **writing your initials** on each page where you have made no other mark.
11. A candidate can be penalized if he/she clearly contradicts him/herself within an answer. Once again make a comment to this effect in the left hand margin.

## Subject Details:      Computer Science HL Paper 2 Markscheme

### Mark Allocation

Candidates are required to answer ALL questions *[20 marks]* for question 1, *[20 marks]* for question 2, *[20 marks]* for question 3 and *[40 marks]* for question 4. 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.
- The order of statements does not have to be as written (unless stated otherwise).
- 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 penalising 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**”.

1. (a) *Award up to [3 marks max] as follows.*

*[1 mark] for inclusion of head.*

*[2 marks] for correct sequence.*

*[1 mark] for sequence with correct head but errors elsewhere.*

The add methods insert at the head.

The correct structure is:

head => 221 => 3 => 1 => 21 => null.

*[3 marks]*

- (b)  $O(1)$ ; (*i.e.* not proportional to the length of the list.)

*[1 mark]*

- (c)  $O(N)$ ; (*i.e.* proportional to the length of the list.)

*[1 mark]*

- (d) Data structure is a dynamic one and uses a chain;  
*i.e.* pointers (Java references) to create the chain, there is no way to reference the address of an element within the list directly other than chaining through the list from the start;

*[2 marks]*

- (e) Recursion;

*[1 mark]*

- (f) *An explanation is required not just a trace. Award [4 marks] for a correct answer, and award [2 marks] for a reasonable attempt at a trace.*

Call to method count executes a call to the countUp method and passes the address of head and assigns to variable n;

Recursively return 1 + call to method countUp again passing next;

This recursive calling process continues until n points to null;

Call sequence terminates and the sequence of 1 + 1 + 1 + 1 is evaluated and returned to the original calling statement;

*[4 marks]*

- (g) *Accept either iterative or recursive solutions. Award up to [4 marks max]. One example of each is listed below for guidance.*

*Possible iterative solution:*

```
boolean search(int data)
```

*[1 mark]*

```
{ Node n = head;
```

```
    while(n != null)
```

*[1 mark]*

```
    { if(n.d == data) return true;
```

*[1 mark]*

```
        else n = n.next;
```

```
    }
```

```
    return false;
```

*[1 mark]*

```
}
```

*continued...*

*Question 1.(g) continued*

*Possible recursive solution:*

```

boolean search(int s)
{
    return searchList(s, head);
}
private Boolean searchList(int s, Node n)
{
    if (n == null) return false;
    if (n.d == s) return true;
    else searchList(s, n.next);
}
    
```

*[1 mark]*

*[1 mark]*

*[1 mark]*

*[1 mark]*

*[4 marks]*

(h)  $O(\log_2 n)$ ;

*[1 mark]*

(i) Root inspected to see if tree empty;  
 If not empty check left or right  
 Inspect node, if match terminate and return found;  
 Else move left or right  
 Terminate when no left or right move option left;

*[3 marks]*

**Total: [20 marks]**

2. (a) (i) *Award [1 mark] for correct answer only.*

–64	32	16	8	4	2	1
0	1	0	0	1	0	1

[1 mark]

- (ii) *Award [2 marks] for fully correct answer. Award [1 mark] if MSB = 1 but error elsewhere.*

–64	32	16	8	4	2	1
1	0	1	1	0	1	1

[2 marks]

- (b) (i) *Award [1 mark].*  
 $0111111 = 63$

[1 mark]

- (ii) *Award [2 marks] for correct answer only. Award [1 mark] for negative but incorrect magnitude.*  
 $1000000 = -64$

[2 marks]

- (c) The digits of the number could not be stored;  
 Creating an overflow error;  
 Which would cause a terminating error unless overflow only into the 7<sup>th</sup> bit;

[3 marks]

- (d) *Award [1 mark] for correct position of point, and [1 mark] for correct bit order.*  
 $4.5 = 100.1$

[2 marks]

- (e) Mantissa = 1001;  
 Exponent = 11;

[2 marks]

- (f) *A base 10 answer is required for full marks.*  
 Exponent =  $3_{10}$  / hence the binary value = 110.1;  
 Answer =  $6.5_{10}$ ;  
*Award [2 marks] for correct answer with no working shown.*

[2 marks]

- (g) *Award up to [2 marks max].*  
 Overflow;  
 Would occur because the number is larger than the largest value that can be presented;  
 Truncation error;  
 Would occur because there are not sufficient bits to store the mantissa;

[2 marks]

- (h) *Award [1 mark] for any reasonable example, [1 mark] for some outline and [1 mark] for some explanation up to [3 marks max].*

*Example:*

Money values;

Need to store fraction amounts e.g. \$23.45;

Errors of accuracy would result if the values could not be stored correctly;

[3 marks]

**Total: [20 marks]**



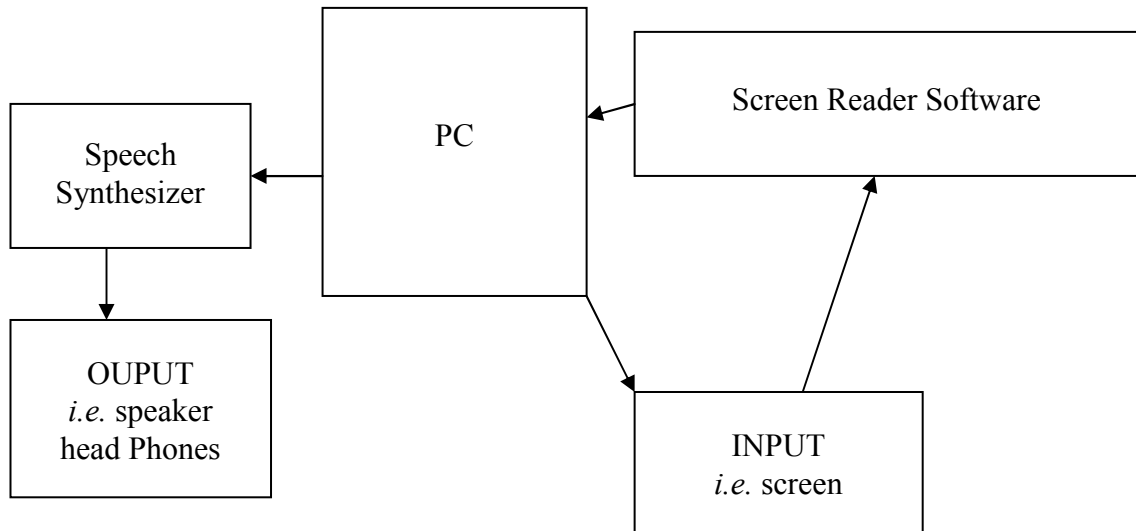
3. (a) Records are organized in a sequence;  
One after the other, the records do not need to be organized in a specified order; **[2 marks]**
- (b) Account name and telephone number grouped in a record;  
Records stored in sequence of account name; **[2 marks]**
- (c) File opened and records read sequentially;  
Account name records checked and if match, telephone number returned; **[2 marks]**
- (d) Data records read from the existing file and written to a new file;  
The new record written to the new file;  
Remaining records read from existing file and sequentially written to the new file; **[3 marks]**
- (e) Additions to maintain order slow;  
And require creating a new file;  
Deletions require location and then creating of a new file;
- or*
- Processing slow to add sequentially;  
And to delete needs sequential search;  
And requires new file to be written each time; **[3 marks]**
- (f) Random organisation / direct/partially indexed / fully indexed;  
Would not require the records to be kept in order;  
And the existing file only is required;  
Deleted records, space could be re-used thus saving space; **[4 marks]**
- (g) *Award up to [4 marks max].*  
Data records could be retrieved via the use of key *e.g.* via hashing, linked to the relative record position **[2 marks]**  
or via an index that relates either the name and/or phone number to the relative record position **[2 marks]**  
or sequentially by reading through the records from start to finish **[1 mark]**. **[4 marks]**

**Total: [20 marks]**

4. (a) Keys small;  
Difficult to handle and/or see;  
  
Layout of keyboard different;  
Unable to see or use efficiently; **[4 marks]**
- (b) Poor eye-sight;  
Makes it difficult to distinguish between graphics and words and directions;  
Colour blind people;  
With otherwise normal vision find combinations of some colours *e.g.* red and green difficult to distinguish; **[4 marks]**
- (c) Two memory arrays;  
One to store the abbreviation;  
And one to store the phrase;  
And linked via a common index;  
  
*or*  
  
A list structure *e.g.* binary tree;  
Storing a record;  
That combines;  
The abbreviation and the phrase into one node of the list; **[4 marks]**
- (d) *Award [1 mark] for method and [1 mark] for brief explanation up to [2 marks max].*  
Array could use hashing;  
To convert abbreviation into index;  
  
*or*  
  
List could be a binary tree;  
And abbreviation searched efficiently using binary efficiency; **[2 marks]**
- (e) *Award [1 mark] for method and [1 mark] for brief explanation up to [2 marks max].*  
Using a pair of arrays the abbreviation index is located;  
And this is used to access the corresponding phrase;  
  
*or*  
  
Using a binary tree the abbreviation is searched for;  
And then the phrase displayed when a match on the abbreviation is found; **[2 marks]**

- (f) *Candidates will draw a range of diagrams. Sample diagram shown.  
Award [1 mark] for each component as follows.*

The input from the screen;  
Screen reader software;  
Speech synthesizer;  
And output device;



[4 marks]

- (g) *Candidates will provide a number of possibilities, one simple one is as follows.  
(Assumes sound of words is stored.)*

Elements of text retrieved;  
Matched to word in sound file;  
Sound of word retrieved;  
Sound played in sequence;

[4 marks]

- (h) *Award [1 mark] for an issue and [1 mark] for an explanation, i.e.  
[2 marks] per example × 3 up to [6 marks max].*

Inability to communicate using online facilities;  
e.g. chat;

Inability to access information using a browser;  
e.g. government websites;

Inability to access entertainment online;  
e.g. (any suitable example);

[6 marks]

**Total: [30 marks]**