

MARKSCHEME

May 2006

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 IBCA.*

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 IBCA. 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 part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each marking 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 mark scheme 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. Effective communication is more important than grammatical niceties.
- 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 [1 mark] for each correct row.

data	hi	lo
37,11,52,7	3	0
37,52,11,7	3	1
37,52,11,7	3	2
52,37,11,7	2	0
52,37,11,7	2	1
52,37,11,7	1	0

[5 marks]

(b) For example

[6 marks]

Award [1 mark] for each of the following comments.

```
int[] arrayjoin(int[] fst, int[] snd){
    int i;
    int largearray[] = new int[fst.length + snd.length]; // sig + vars
    for(i = 0; i <= largearray.length - 1; i++){ // loop
        if ( i< fst.length - 1){ // test
            largearray[i] = fst[i]; // transfer fst
        } else {
            largearray[i] = snd[i - fst.length - 1]; // transfer snd
        }
    }
    proc(largearray, 0, largearray.length - 1) // sort
    return largearray
}
```

Alternatively

```
private static int[] arrayjoin (int[] data1, int[] data2)
{
    int[] res = new int[data1.length + data2.length]; // sig + vars
    for (int i = 0; i < data1.length; i++) // loop
    { res[i] = data1 [i]; // transfer data1
    }
    for (int i = 0; i < data2.length; i++) // loop
    { res [data1.length + i] = data2[i]; // transfer data2
    }
    proc ( res, 0, res.length-1); // sort
    return res;
}
```

In either case award marks as indicated in the comments at the side.

(c) For example

[6 marks]

Award [1 mark] for each of the following comments.

```
private static int[] arraydups (int[] data)
{
    int nDups = 0; // sig + vars
    for (int i = 1; < data.length; i++) // loop
    { if (data [i-1] ==data[i]) nDups++; // test
    }
    int[] res = new int[data.length - nDups]; // create array of
                                                // correct length

    res[0] = data[0];
    int j = 1;
    for (int i =1; i < data.length; i++)
    { if (data[i-1] !=data[i]) // test
      { res[j] =data[i]; // transfer only
        // non-duplicates

        j++;
      }
    }
    return res;
}
```

(d) *Award [1 mark] for each of the following comments.*

[3 marks]

```
void msort(int[] fst, int[] snd){
    int[] largearray = new int[fst.length + snd.length] // sig + vars
    lgarray = arrayjoin(fst, snd) // join
    arraydups(lgarray) // remove dups
}
```

- 2. (a) *Award [1 mark] for each process.*

For example

security process applied;
 address added;
 re-assembly data added;
 check data added;

[4 marks]

- (b) *Award [1 mark] for each step.*

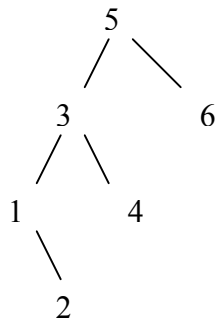
For example

each packet travel separately;
 routed at nodes;
 according to address;
 and route availability;

[4 marks]

- (c) *Award up to [4 marks] for outline. [1 mark] for each following point:*

packets checked using check data;
 if incorrect re-transmit requested;
 binary tree populated;
 according to assembly data;
 security process applied;



[6 marks]

Award [2 marks] all correct, [1 mark] with 1 mistake.

- (d) *Accept any reasonable method justified.*

For example:

Fibre optic;
 Fast transfer, less time to corrupt **[2 marks]**.

[3 marks]

- (e) Data may be intercepted and read or changed;
 Encrypt the data so that it is unintelligible to anyone not supposed to read it;
 Give the key to decrypt to the intended receiver **[2 marks]**.

[3 marks]

3. (a) sequential;
partially indexed;
fully indexed;
direct access; **[4 marks]**
- (b) sequential;
data always appended;
always accessed sequentially;
order is part of embedded information; **[4 marks]**
- (c) For example:

fully indexed;
unordered required as files too large to move around;
yet fast access can be achieved via index;
as index can be sorted;
and binary search used to locate;
followed by direct access; **[6 marks]**
- (d) redundant data is removed from the file; **[1 mark]**
Do not award a mark for the file is made smaller only.
- (d) For example:

tape;
as can be very large;
portable;
therefore can be stored remotely;
and fast access is not required; **[5 marks]**
Accept more modern and likely storage systems with [1 mark] for medium and [4 marks] for a good explanation.

4. (a) *Award [1 mark] for each point.*
Sampling records the wave shape;
of the music by measuring its amplitude;
at short given intervals;
assigning a numerical value to it;
it is these numbers that are saved and played back; **[5 marks]**
- (b) *Award [1 mark] for each point.*
MIDI events are the recorded actions;
from instrument with MIDI converter;
or computer input;
such as note, volume, duration etc;
that were used to create the notes;
numerical values are given to each aspect of the event;
and used to recreate the music by recreating the actions;
via a MIDI device; **[8 marks]**
- (c) *Award [1 mark] for each point.*
MIDI for jingle;
small file size;
easy to edit;
sampling for school;
can handle singing;
file size not important; **[6 marks]**
- (d) *Award [1 mark] for method and two for explanation e.g.*
a collection of integers to store the data;
grouped in a linked list;
data is positive unsigned no float,
string harder to manipulate;
linked list has undefined length;
and easy to edit; **[6 marks]**
- (e) *Award [1 mark] for each point.*
various input devices;
including non-musical;
can be used to create MIDI data;
that can be manipulated to create many musical parts;
that can be played without the need for a band; **[5 marks]**
- (f) Multi-timbral **[1 mark]**: this feature allows a receiving device / sound module / synthesiser to receive and play data from more than 1 channel **[1 mark]**.
Polyphonic **[1 mark]**: this feature allows a receiving device / sound module / synthesiser to play more than one note at a time **[1 mark]**. **[4 marks max]**

- (g) (i) Channel 4; **[1 mark]**
- (ii) by looking at the MSB (allow the leftmost bit, but not the first bit) **[1 mark]**, 1 for a status byte and 0 for a data byte **[1 mark]**. **[2 marks]**
- (iii) 128; **[1 mark]**
- (iv) Award **[1 mark]** for the first byte and **[1 mark]** for the third byte.
10011011 00111100 01111111 **[2 marks]**
-