INTERNATIONAL BACCALAUREATE
BACCALAURÉAT INTERNATIONAL
BACHILLERATO INTERNACIONAL

M02/650/H(2)M+

MARKSCHEME

May 2002

COMPUTER SCIENCE

Higher Level

Paper 2

13 pages

Subject Details: Computer Science HL Paper 2 Markscheme

Mark Allocation

Candidates are required to answer ALL questions (***[30 marks]*** for question 1, ***[30 marks]*** for question 2 and ***[15 marks]*** for the remaining three questions). Maximum total = ***[105 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.
- The order of points 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 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 penalised. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with **"FT"**.

1. (a) 56 minutes *(accept 57 minutes)*; ***[1 mark]***
*(Do **not** accept "cannot tell")*

(b) *Award **[1 mark]** for each valid point, up to a maximum of **[3 marks]**.*
an array is fixed in size;
whereas a list is not (/limited only by size of RAM);
or an array is a static data structure and a list is a dynamic data structure ***[2 marks]***;
the number of readings is not fixed;
so it would be difficult to state at the start of the program;
or an array would have to be set at 1440 (24 × 60) elements / which is a waste of space. ***[3 marks]***

(c) *Accept by written description or an annotated diagram, awarding marks for the following points shown / outlined, up to a maximum of **[4 marks]**:*
record structure;
three (separate) fields;
fields generally appropriately named (*e.g.* temp(erature), time, link / next / pointer *etc.*);
suitable data types for each field, *e.g.* integer / real for temperature, pointer for the link, and for time integer. ***[4 marks]***
*(**[1 mark]** for at least 2 appropriate fields).*
*Give an additional mark (so long as it does not make total go over **[4 marks]**) for stating "start" will be of type pointer.*

(d) For example:

```
function  GETTIME(val TEMPERATURE: integer): integer
   declare CURRENT pointer --> NODE
   declare PREV pointer --> NODE
   declare TOTAL:integer
   CURRENT <-- START
   TOTAL <-- 0
   while CURRENT # nil do
      PREV <-- CURRENT
      CURRENT <-- CURRENT --> NEXT
      if CURRENT --> TEMP = TEMPERATURE then
         TOTAL <-- TOTAL + (CURRENT --> TIME) – (PREV --> TIME)
      endif
   endwhile
   GETTIME <-- TOTAL
endprocedure GETTIME
```

Award marks as follows:
correct function with `TEMP` as **`val`** parameter;
declaration of **most** variables;
correct loop through list:
  current set to `START`;
  correct conditions for loop;
  change of pointers;
correct test for temperature;
correct calculation of `TOTAL` ***[2 marks]****. Award **[1 mark]** for a good attempt e.g. if previous temperature is not subtracted or if total has not been set to 0 initially)*
Assign `TOTAL` to `GETTIME`; ***[9 marks]***

(e) For example:

```
procedure MIN_MAX (ref MIN, MAX: integer)
     declare CURRENT pointer --> NODE
     CURRENT <-- START
     MAX <-- 0
     MIN <-- 100
     while CURRENT # nil do
       if CURRENT --> TEMP < MIN then
          MIN <-- CURRENT --> TEMP
       else if CURRENT --> TEMP > MAX then
          MAX <-- CURRENT --> TEMP
       endif
       CURRENT <-- CURRENT --> NEXT
     endwhile
endprocedure MIN_MAX
```

Award marks as follows:
correct parameters passed;
variables declared;
setting MIN and MAX appropriately ***[2 marks]***.
loop through list;
test and assign MIN;
test and assign MAX; ***[7 marks]***
Accept any reasonable initial values for MIN *and* MAX.

(f) For example:

```
procedure CREATE_TABLE (val MIN, MAX : integer, ref TEMP array of
integer)
declare P, T, C integer
P <-- 1
for C <-- MIN to MAX do
    T <-- GETTIME(C)
    if T > 0 then
       TEMP[P,1] <-- C
       TEMP[P,2] <-- T
       P <-- P + 1
    endif
enddo
endprocedure CREATE_TABLE
```

Award marks as follows:
passing of parameters as correct type;
declaration of most variables;
loop from MIN to MAX ***[2 marks]***. *Award **[1 mark]** for a reasonable attempt.*
check for time > 0;
transfer to array ***[2 marks]***. *Award **[1 mark]** if incorrect attempt made.* ***[6 marks]***

2. (a) *Award **[1 mark]** for need and **[1 mark]** for reason, up to a maximum of **[4 marks]**.*
high number of computations;
needs fast processor;
vast input / output;
needs large RAM;
size of scans;
requires large secondary storage; ***[4 marks]***

(b) (i) 2^8 **or** 256 shades of grey; ***[1 mark]***

(ii) *Award marks as allocated up to a maximum of **[2 marks]**.*
256 × 512 bits × 512 bits;
= 256 Kb; ***[2 marks]***

(c) *Award marks as allocated, up to a maximum of **[6 marks]**.*
abuse *definition **[1 mark]***;
crime definition ***[1 mark]***;
state example of abuse ***[1 mark]***, further explanation ***[1 mark]***;
state example of crime ***[1 mark]***; further explanation ***[1 mark]***;
For example:
Computer crime is the use of the computer for illegal purposes, e.g. taking pictures of CT scans...
Computer abuse is the use of computer technology/data for purposes other than that for which it was intended, e.g. hacking into... ***[6 marks]***
The two examples given are two possible answers but accept any that is valid and appropriate to the Case Study.

(d) *Award marks as follows:*
definition of data compression ***[2 marks]***;
to make smaller ***[1 mark]*** can be restored ***[1 mark]***
*For each example **[1 mark]** for a valid example and **[1 mark]** for further explanation.*

Accept any reasonable example from the Case Study. ***[6 marks]***

(e) *Award marks as allocated, up to a maximum of **[8 marks]**. Diagrams will vary but the following links required (two example diagrams are included).*
***[4 marks]** for communications.*
Internet;
WAN;
2 LANs;
medical lab;

***[2 marks]** for*
input / output from medical centre;
Any two peripherals at centre accepted.

***[2 marks]** for*
clear set up of the 2 LANs;
5 workstations and hub ***[1 mark]***;
connection between ***[1 mark]***; ***[8 marks]***

(f) *Award* ***[1 mark]*** *for identifying a major advance and* ***[1 mark]*** *for discussion / explanation, up to a maximum of* ***[2 marks]***. ***[3 marks]***

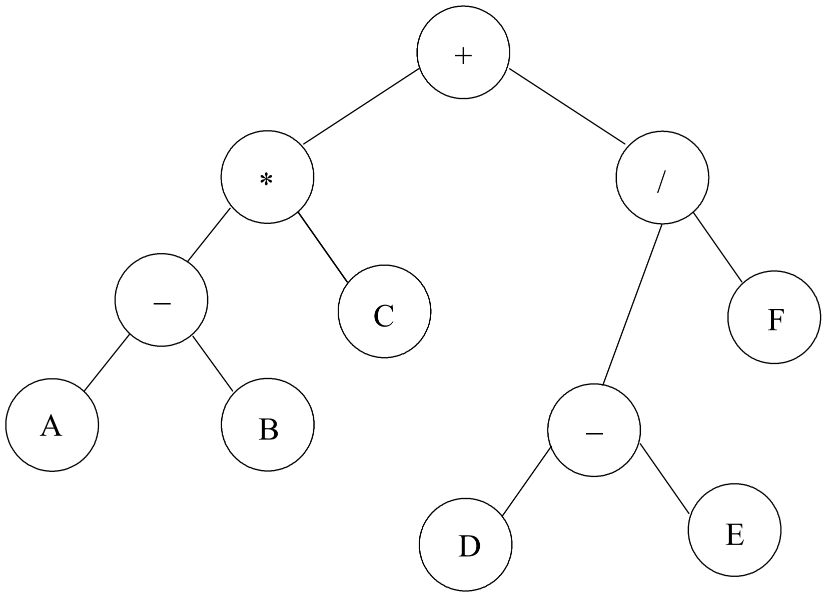
e.g. Virtual Anthropology using to CT scans is one advance: helps the dating of fossils and hence we know more about our history;
Use of scans in medicine means diagnosis is more accurate and hence more effective treatment can be given;

Accept advances such as faster processors, larger storage etc. as long as these are then related to an impact on scientific research.

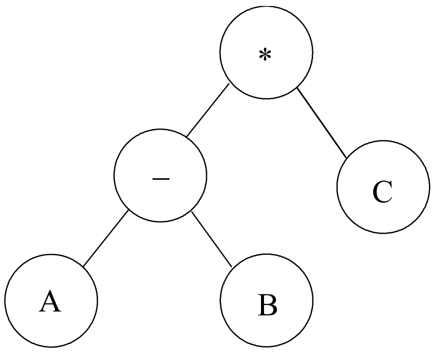
3. (a) (i) ABC+*D–; ***[1 mark]***

(ii) A*B+C–D; ***[1 mark]***
Do not accept A(B+C)-D, i.e. no brackets.*

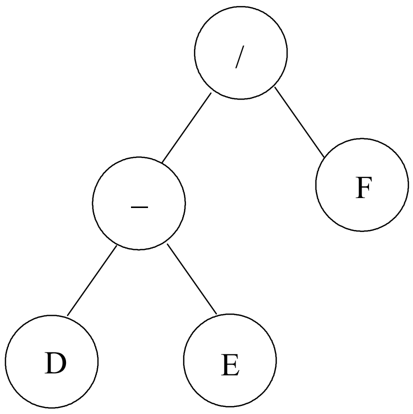
(b) The tree is:



Award marks as follows:
[1 mark] for + as the root
[1 mark] for the subtrees A-B **and** D-E (regardless of position)
[1 mark] for the correct subtree



(in any position)
[1 mark] for the correct subtree



(in any position) ***[4 marks]***

(c) *Award marks as follows (which can be gained by description as text, or a clear, annotated diagram) up to a maximum of* ***[3 marks]****:*
a record;
(at least) three fields;
two of the fields named left (link / pointer) / right (link / pointer) *etc.*;
one field to hold operator / operand ***[3 marks]***

(d) (i) Algorithm is:

```
procedure POST_ORDER(val ROOT pointer->NODE)
   if ROOT # nil then
      POST_ORDER(ROOT->LEFT)
      POST_ORDER(ROOT->RIGHT)
      output ROOT -> DATA
endif
```

Award marks as follows, up to a maximum of ***[5 marks]****:*
for a parameter related to `ROOT` or `NODE` *etc.* (no data type is required);
for test to terminate procedure, (similar to # nil, can be done before recursive calls);
for a good attempt at recursion using links (*e.g.* `LEFT,RIGHT` *etc.*, not just **any** recursive call, *e.g.* no marks for `POST_ORDER(ROOT)` *i.e.* **same** parameter);
for left call before right call;
for output at the end; ***[5 marks]***

(ii) for adapting the algorithm given by the candidate in (d)(i) to order output, left, right, *e.g.*:

```
procedure PRE_ORDER(val ROOT pointer->NODE)
   if ROOT # nil then
      output ROOT -> DATA
      PRE_ORDER(ROOT->LEFT)
      PRE_ORDER(ROOT->RIGHT)
   endif
```

[1 mark]

4. (a) Fetch; ***[1 mark]***

(b) (i) Virtual memory; ***[1 mark]***

(ii) *Give **[1 mark]** for a correct aspect (up to **[2 marks]** maximum), and **[2 marks]** for a valid **comparison and reason** (i.e. not just a statement like "this is faster in RAM than disk") (**[1 mark]** for a partial comparison).*
latency or rotational delay
waiting for the block / location to reach / pass under head which is not required in RAM, so faster in RAM;
seek time
the head needs to be moved to the correct track on disk, which is much slower than locating a memory location in RAM;
transfer time
the transfer time is dependent on the speed of rotation of the disk, which is slower than the speed of data in the CPU / along the internal bus; ***[6 marks]***

(iii) *Allocate marks as follows, up to a maximum of **[3 marks]**.*
hash algorithms use a calculation (to generate the address);
which is faster (than searching an index);
and speed is important (for virtual memory); ***[3 marks]***

(c) *Allocate **[1 mark]** for a valid OS function and **[1 mark]** for a further elaboration, up to a maximum of **[4 marks]**.*
file maintenance;
such as deleting / renaming *etc.*;
software execution control;
identifying which (part of which) program should be executed (at any point);
security;
such as allowing access to network / files using passwords *etc.*;
Accept any further finctions of an operating system. For example:
provide a user interface;
GUI or CLI
error handling; such as runtime errors;
interface with peripherals; or I/O control
sending/receiving/checking; data ***[4 marks]***

5. (a) Advantage: the data collected is real;
Disadvantage: the current user may "perform" for the analyst; ***[2 marks]***

(b) Advantage: large pool of data;
Disadvantage: responses may be ill considered; ***[2 marks]***

(c) *Award **[1 mark]** for a feature, and up to **[1 mark]** for an explanation.*
For example:
Feasibility report (accept a named report which has the same meaning)
cost / money
the new system may require less staff, so salary outgoings are less thus saving money;
the new system may produce more goods (relevant output) in the same time, so that sales can increase and produce more income *etc.*;
speed
the new system may produce results faster than the old system;
quality / accuracy
the new system may produce more detailed / accurate results than the old system which may improve actions;
the new system may have less crashes / be more robust, thus allowing more work *etc.*;
a description of the current system;
an outline of the proposed system;
implication for workers;
time needed to changeover; ***[3 marks]***

(d) *Accept any reasonable explanation that relates to factory.*
For example:
Input could be by touch screen or touch pad to avoid clogging the keyboard;
Output could be by LCD or sound (easily seen/heard) in workshop;
Or paper tape as it is transferred to machine and if damaged this is obvious;
Storage on punched card/tape for transfer to machine;
Processor less powerful and still meet specification (cheaper or more expendible);
Communications links need to be robust therefore invest in better cabling; ***[8 marks]***