

# **MARKSCHEME**

**November 2000**

**COMPUTER SCIENCE**

**Higher Level**

**Paper 1**

**SECTION A**

1. (Award [**1 mark**] for the idea that it (is hardware/software combination that) connects networks together; idea that it directs data to the appropriate path.)
  
2. (Award [**1 mark**] for temporary; and [**1 mark**] for store/memory.)
  
3. (a) (Award [**1 mark**] for the idea that it's a data-structure whose items are added to the rear of a queue/one end; and [**1 mark**] that items are removed from the front/other end.)

Give [**2 marks**] for stating that it is a First-in, First-out structure (or Last-in, Last-out). Give NO marks for just stating FIFO structure!  
DO NOT accept any facile comments about supermarket queues *etc.* – this is a Higher Level Computer Science exam!

- (b) (Award [**1 mark**] for a valid application, and a second mark for an attempt at a valid description:)
  - Jobs waiting to be executed by a computer, a new job is added to the end, computer executes next job from front;
  - Keyboard buffer storing data whilst the processor is busy doing another task;
  - Spooling output to a disk to await printing, new jobs added at the end, printer deals with jobs from front/accept idea as a print queue in general.

4. (Award [**1 mark**] for  $O(n \log n)$ )

5. (a) Two (or more) different values can give the same result from the hash algorithm. [**1 mark**]
- (b) (Award [**1 mark**] for a valid method, and a second mark for an attempt at a valid description:)
  - Locate next free space and store data item there;
  - Have an overflow area with a marker/pointer to it.

6. (Award [**1 mark**] for a correct stage, and a second mark for a correct elaboration, up to a maximum of [**6 marks**].)
  - Systems analysis, an investigation which leads to a precise statement of the problem;
  - Software/program design, a breakdown of the problem statement into its constituent parts from which coding can take place;
  - installation/operation, the introduction of the system so that it can be used by the end-user;
  - maintenance, where the system is checked for errors/improvements which will lead to another cycle.

7. (Award the marks as indicated below; up to **[4 marks]** max:)

- A function should return one value;
- which is returned by the function name/itself;
- parameters should not change/no 'side-effects';
- since this would mean more than one value is returned;
- so there is no need for pass-by-reference parameters;
- which can be changed;
- unlike pass-by-value parameters (which can't be altered);
- so pass-by-value parameters should be used;
- unless pass-by-reference parameters are used to save memory;
- and the values are not changed.

8. (Award up to **[2 marks]** for an outline of encapsulation and up to **[2 marks]** for an outline of polymorphism:)

Encapsulation: (Award **[1 mark]** for each of the points indicated below; up to **[2 marks]** max:)

- the combination of data and the operations that act on the data;
- into a single unit/object;
- allowing information/data hiding.

Polymorphism: (Award **[1 mark]** for each of the points indicated below; up to **[2 marks]** max:)

- The same operation can be applied to different objects;
- and the object behaves appropriately/'differently';
- allowing simpler/generic code.

9. (Award **[1 mark]** for identifying a valid benefit, and up to **[2 marks]** for a clear explanation, for two benefits, giving **[6 marks]** max:)

- many users can access the data at the same time;
- this means that users can view the data when they want;
- without having to wait if someone else is using it;
- without having to move to a central file/computer;
- because they can access it from their own terminals.

(Note the above can be separated into two points, i.e. 'many users' and 'many terminals/locations' - this is acceptable for two separate advantages.)

- data integrity is easier to maintain;
- with only one copy of the data, rather than many copies;
- any changes are recorded in the central database;
- meaning that there are not separate files with different data as individual ones are updated.

10.

A	B	AxorB
0	0	0
0	1	1
1	0	1
1	1	0

[1 mark]

[1 mark]

[1 mark]

[1 mark]

$$= \bar{A}.B + A.\bar{B} \quad [1 \text{ mark}]$$

$$(A + B). (\overline{A.B})$$

$$(A + B). A$$

11. (Award [1 mark] for identifying a suitable advantage other than speed-related, and [1 mark] for a further correct elaboration, and [1 mark] for identifying a suitable disadvantage, and [1 mark] for a further correct elaboration, up to a max of [4 marks].)

ADVANTAGES:

- Security:
  - email will only deliver to the specified address (whereas normal mail could be opened by another person); or
  - email addresses usually require a password to access it (whereas physical mail can be opened by another person);
- Economy:
  - in most countries the cost of a local call is cheaper than the international mail rate;
- Convenience:
  - the mail can be sent without having to move from the computer (unlike a letter which needs to be packaged, weighed, correct stamps bought *etc.*);

Do NOT accept:

- Multiple sendings: the same email can be sent to a group of people. (So can a document, *i.e.* photocopy it!) This idea CAN be accepted IF the candidate explains that it would save the **inconvenience** of photocopying *etc.*, because then it's the previous point!
- Attach and send replies *etc.* because this can be done with physical documents; *i.e.* don't accept tasks that are equally valid with paper documents.

DISADVANTAGES:

- The original document is not received:
  - this may be required in some cases (*e.g.* legal contracts);
- No physical items can be included:
  - additional articles cannot be included such as a product sample (or even separate handwritten notes *etc.* - see next point);
- Personalised notes may be lost:
  - although notes *etc.* can be scanned and so the original layout/format/colour maintained, this is more difficult than simply enclosing original notes/letters and so personal comments/intimations may be lost. (Accept the more concrete "this cannot be done" from a candidate, as well as the correct "more difficult").

**SECTION B**

12. (a) Boolean. *[1 mark]*

(b)

HALF	MIDDLE	POSITION	SAME	COUNT	
3	4	1	true	1	<i>[1 mark]</i>
		2	false		<i>[1 mark]</i>
		3	true	2	<i>[1 mark]</i>

(c) (Award *[2 marks]* for a complete explanation, *[1 mark]* for a partial answer.)

Complete answers:

It counts the number of values that are equal *[1 mark]* at equivalent (opposite) locations from the centre *[1 mark]*.

It tests matching entries from the centre *[1 mark]*, counting how many are equal *[1 mark]*.

It tests symmetrical/balancing locations *[1 mark]*, seeing how many are equal *[1 mark]*.

It counts the number of entries which are the same *[1 mark]* mirrored about the centre/middle (of the array) *[1 mark]* etc.

Partial answers:

It counts how many entries make it a palindrome *[1 mark]*.

It tests if it is a palindrome *[1 mark]*.

It counts if the ends are equal *[1 mark]*.

It looks as if it is a mirror *[1 mark]* etc.

(d) (Award marks as follows:)

- *[1 mark]* for stating that COUNT changes within the procedure;
- *[1 mark]* for the idea that it needs to be passed back to (or ‘used’ by) the calling routine/main program.

(e) (Award marks as follows, up to a maximum of *[2 marks]*:)

- *[1 mark]* for identifying that a function returns a single value;
- *[1 mark]* for stating that since this is what the algorithm does it is appropriate;
- *[1 mark]* since there is only one ‘out’ parameter;
- *[1 mark]* and the others are ‘in’ parameters;
- *[1 mark]* the value can be passed back via a function name.

(Check other apparently correct answers with your team leader.)

13. (a)

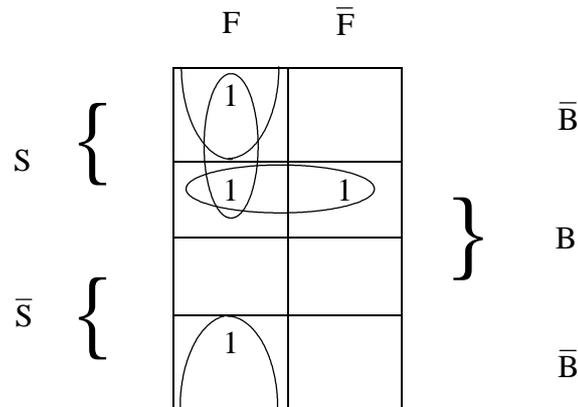
F	S	B	N
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

[1 mark] for all 8 inputs.

[3 marks] for all N correct (subtract [1 mark] for each incorrect value, do **not** award negative marks!)

(b)  $N = \bar{F}SB + F\bar{S}\bar{B} + FS\bar{B} + FSB$  [1 mark]  
(or  $SB + FS + \bar{F}B$ )

(c) Using a Karnaugh Map:



$$N = SB + F\bar{B}$$

By Boolean Algebra:

$$\bar{F}SB + F\bar{S}\bar{B} + FSB + FSB$$

$$SB(\bar{F}+F) + F\bar{B}(\bar{S}+S)$$

$$= SB + F\bar{B}$$

**[1 mark]** for any simplification by 1 level.

**[2 marks]** for simplifying by 2 or more levels.

**[3 marks]** if final solution uses 4 gates (if only final solution is given and correct, candidate gets all **[3 marks]**).

*N.B:* if candidate does not attempt this part, but gave answer as  $FS + F\bar{S}\bar{B} + \bar{F}SB$  in part (b), give **[1 mark]** here (for implicit simplification by 1 level).

(d) The number of bits available is not enough for answer **[1 mark]** so MSB becomes 1 indicating a negative value **[1 mark]**.

(If candidate shows calculation and a 1 ends up in MSB give **[1 mark]**, with explanation gets second mark.)

14. (a) (i) (Award the marks as follows:)

As the year changed to 2000, the date (NOW) would have been stored as 00; [1 mark]  
so subtracting a value from it would have given a negative value/wrong value [1 mark]

Do NOT accept ‘millennium bug’ without an explanation.

(ii) • (Award [2 marks] for a clear description ([1 mark] for a partial answer):)

The new algorithm will work for any account that is not open for more than 99 years. There is no time limit (*i.e.* it does not stop functioning in 2100).

• (Award a further [1 mark] for showing a correct calculation, and the remaining second mark for explaining how an account 100 years earlier is wrong; OR [2 marks] for explaining how an account opened over 100 years ago is wrong:)

– If NOW was 2101 (*i.e.* 01) and an account was opened in 2099 (*i.e.* 99) answer should be 2. Calculation gives  $01+100-99 = 2$ , correct. But same answer would be given for 1999 instead of 102, *i.e.* wrong.

OR

– If an account was opened in 1950, and NOW is 2051. The calculation would give  $51-50$  which gives 1 year, instead of 101.

(b) (Award [1 mark] for a valid point regarding system documentation, and a second mark for an elaboration, for two points, up to [4 marks] max:)

- Structure diagrams/data flow diagrams *etc.* [1 mark]  
show the logic of the algorithm, so it would be easy to detect where the change is needed. [1 mark]
- An annotated program listing/description [1 mark]  
would guide a programmer to where the required calculation is located to be altered. [1 mark]

15. (a) (Award **[2 marks]** for a complete answer, **[1 mark]** for a partial answer.)

If one sensor malfunctions/breaks down **[1 mark]** it will be detected by comparing with the other two **[1 mark]** (i.e. two will give one reading, the broken one a different reading - this would get the explanatory mark).

“In case one breaks down” gets **[1 mark]**.

“In case it breaks down” gets **[0 marks]**.

DO NOT accept for taking readings at different/three places. (The question states they are at **one** place.)

- (b) **[1 mark]** for:

Analog(ue) to digital conversion. (Do **not** accept just ADC.)

- (c) (Award **[1 mark]** for any of the following points, up to **[3 marks]** max:)

- Each sensor is monitored/read/accessed;
- in turn/on a regular basis;
- The (master) processor requests data in a “round-robin” approach;
- Data is stored from each sensor (for processing);

Accept (for full marks):

- Each sensor stores data in a(n input) buffer/after ADC data is stored in a buffer;
- The processor compares the three readings for equality before starting polling again.

- (d) (Award **[1 mark]** for each point where a comparison is valid, and a further mark for an elaboration, for two separate points, giving a maximum of **[4 marks]**.)

- Speed of data flow **[1 mark]**
- Because bits are sent simultaneously/at the same time in parallel, it is faster than serial where they are sent one after the other. **[1 mark]**

“Parallel connection is faster than serial connection” is worth **[1 mark]** for the valid comparison point, but no more.

“Parallel is faster because the bits are sent at the same time” is worth **[1 mark]**.

“Parallel is faster because the bits are sent at the same time but serial is one after the other” is worth **[2 marks]**.

- Links/lines/‘cables’ **[1 mark]**

Only one line is required in serial transmission (with bits sent consecutively) where many lines are required in parallel (due to simultaneous bit transmission). **[1 mark]**

16. (a) (Award [**1 mark**] for any of the following points up to [**3 marks**] max:)

(NOTE: The question asks for WHY defragmentation is required NOT how!)

- As data/files are deleted and added;
- contiguous data is not physically next to each other on the disk;
- hence the disk head has to move much more to retrieve data;
- which slows down the (reading) process.

DO NOT accept answers relating to movement of records within files.

(b) (Award [**3 marks**] for a clear description of timesharing as follows (just stating 'timesharing' or 'multitasking' gets [**1 mark**]), up to [**3 marks**] max:)

- Using timesharing/multitasking [**1 mark**];
- processor time is divided between the two programs [**1 mark**];
- each program is given a set timeslice/period of time [**1 mark**];
- after which the other program is given a timeslice, and this process is repeated [**1 mark**];
- the switching between programs occurs so fast [**1 mark**] that to the user it appears that both are running simultaneously/at the same time.

(c) (Award [**1 mark**] for each of the following points, up to [**4 marks**] max:)

- Each email message has the address of the receiver (and of the sender);
  - The message is sent around the LAN (in 'packet/s');
  - The server stores the message;
  - and sends a message to the secretary that email has been received;
-