INTERNATIONAL
BACCALAURÉAT
BACHILLERATO

IB

BACCALAUREATE
INTERNATIONAL
INTERNACIONAL

MARKSCHEME

November 2000

COMPUTER SCIENCE

Standard Level

Paper 1

12 pages

SECTION A

1. *Syntax* = mistake in the use of programming language *[1 mark]*.
   Logical = use of language OK but result not intended result *[1 mark]*.
   Run-time = error which only appears during operation *[1 mark]*.

2. Cheap *[1 mark]* <u>OR</u> large *[1 mark]*
   Non-volatile *[1 mark]*.

3. Verification *[1 mark]*

4. Results are generated fast enough (accept "immediately") *[1 mark]* to influence the next
   input/process *[1 mark]*.
   Do not accept 'on-line'.

5. 1 *[1 mark]*

6. Any two from

   – While (...do)
   – Repeat (...until)
   – For (...do)

   for *[1 mark]* each up to a maximum of *[2 marks]*.

7. – Coordinates *[1 mark]* the execution of the software *[1 mark]*
   – Carries out *[1 mark]* the fetch execute cycle *[1 mark]*

8. *(Award [1 mark] for the idea that it (is hardware/software combination that) connects
   networks together; idea that it directs date to the appropriate path.)*

9. *(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;
   – documentation.

10. *(Award marks as follows, up to [2 marks] maximum:)*

– When sending data over a network *[1 mark]* if a validation/error check detects an error *[1 mark].*
– A transmission check, *e.g.* parity, comparison of double send *[1 mark]* may find an error *[1 mark].*

11. *(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.

12. *(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.*);
    – can send pictures and sound in computer readable format.

Do <u>NOT</u> accept:
–   Multiple sendings: the same email can be sent to a group of people. (So can a document, *i.e.* photocopy it!) This idea <u>CAN</u> be accepted <u>IF</u> 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 <u>original</u> 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")'
– Can mistakenly send viruses:
    – examples of damage caused by viruses.

SECTION B

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

(b)

HALF	MIDDLE	POSITION	SAME	COUNT	
3	4	1	true	1	*[1 mark]*
		2	false		*[1 mark]*
		3	true	2	*[1 mark]*

(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.)

14. (a) 11111111 *[1 mark]*

256 colours *[1 mark]*
Do NOT award any marks for 255, or 0-255.

(b) 00010100 *[1 mark]*

(Do NOT award any marks for 10100 or 010100 (i.e. it MUST be in 8 bits).)

(c) *(Award [1 mark] for identifying a suitable area of standardisation, and [1 mark] for a correct elaboration, up to [2 marks] max:)*

    – Network data/protocols *[1 mark]*
      – so communication can be "understood" around/between networks *[1 mark]*;
      – so data can be transferred without translation/conversion; *[1 mark]*

    – (Document/graphic) Files *[1 mark]*
      – so different packages can import/export documents without problems/conversion. *[1 mark]*

(d) $30\,000 \times 1230 = 36\,900\,000$ kB

$(1\,GB = 1024 \times 1024 \times 1024$ bytes)

$36\,900\,000 / (1024 \times 1024) = 35.19$ GB

(Award marks as follows:)
 – *[1 mark]* for giving answer as 36.9 GB (The idea of relationship between GB and kB is clear.)

    – *[1 mark]* for ANY attempt at dividing by 1024 instead of 1000 to get final answer.
      – Obviously 35.19 or 35.2 GB is fine (even if 1024 is not seen in the working!);
      – So is 35 GB <u>if</u> 1024 has been shown in working;
      – Accept 36.035 GB without seeing 1024 in the working (this is obtained by initial division of 1024, then by 1000 to get GB);
      – Accept 36 GB <u>if</u> initial division by 1024 is shown on page to get 36035(.156), then directly by 1000.

If answer is left as $\dfrac{36\,900\,000}{1024 \times 1024}$ GB or even $\dfrac{(30\,000 \times 1230)}{1024 \times 1024}$ GB give both marks.

(Note that there are NO marks for the initial working of getting 36 900 000!)

(e) *(Award marks as follows:)*

    – *[1 mark]* for identifying that data compressors reduce storage size (or "saves space");
    – *[2 marks]* for explaining a correct situation:
      – to transfer data between computers *[1 mark]* using a smaller storage medium (*e.g.* floppy disc) *[1 mark]*;
      – to save space on the hard disc *[1 mark]* so that more data/software can be stored *[1 mark]*. (In fact this statement would get all three marks, *i.e.* "save space" "on the hard disc" gets *[2 marks]*, then the third for the reason);
      – quicker transfer of data.

15. (a) *(Award **[1 mark]** for identifying an area where a difference occurs, and **[1 mark]** for a correct elaboration, for 2 different areas, giving a maximum of **[4 marks]**:)*

– Production of code **[1 mark]**: a compiler produces a separate machine code version, unlike an interpreter (which executes as it goes, not saving object code) **[1 mark]**;

– Input requirements **[1 mark]**: a compiler needs a complete HLL program, unlike an interpreter which will start with any code until it runs out of statements/error **[1 mark]**;

– Error handling **[1 mark]**: a (good) compiler will list all errors in the program, an interpreter will stop and report at first error **[1 mark]**;

– Program complexity **[1 mark]**: a compiler is a more complex/"larger" program (because it has more functions *e.g.* optimisation), so requires more memory than an interpreter **[1 mark]**);

– Final program requirements **[1 mark]**: once finished and compiled the compiler is not required again to execute the code, an interpreter is required to be loaded/"used" every time the program is executed **[1 mark]**).

(NOTE: the candidate does not have to specify the area EXPLICITLY as given in the markscheme above. Full marks would be gained by the section after the colon (:) in EACH case, because the area is obvious by the description.)

(b) *(Award **[1 mark]** for a valid item of system documentation and a further **[1 mark]** for a correct statement as to how it would be used in maintenance for two items, giving a maximum of **[4 marks]**.)*

– Program Design/Structure diagrams/Pseudocode **[1 mark]**
– to show logic so easy to see how to debug/modify/update **[1 mark]**

– (Anotated) program listing **[1 mark]**
– so another programmer can use code to debug/modify/update **[1 mark]**

– description of data structures **[1 mark]**
– so another programmer can change them/use them in another procedure/add a new field **[1 mark]**

– test strategy/testing **[1 mark]**
– so another programmer is aware of data types/format that can be used for alteration/update.

(c) *(Award **[1 mark]** for a valid reason, and **[1 mark]** for a correct elaboration:)*

– **[1 mark]** for any mention of a web (or internet) page;
– **[1 mark]** for a reason why, *e.g.* to advertise the software company's services; to create a site for a client.

16. (a) *(Award [1 mark] for a valid ethical issue, and [1 mark] for a suitable elaboration; for two issues, giving a maximum [4 marks]:)*

– software piracy *[1 mark]*, by copying CD-ROMs money is not being given to the proper person (*i.e.* the author) *[1 mark]*;

– defrauding the company *[1 mark]*, the employee is stealing from his employer by not using work time properly. (Accept cost of unauthorised phone calls, even though this may not be directly accurate). *[1 mark]*;

– work spying on employees *[1 mark]*, it may be seen as intimidating that managers can view every email sent/received by each employee *[1 mark]*.

(b) *(Award [1 mark] for a valid precaution, and [1 mark] for a suitable elaboration to give a total of [2 marks]:)*

– constant virus checking software *[1 mark]* to test all incoming emails (and attachments) *[1 mark]* ;

– a firewall *[1 mark]* such as ring back connection so that only authorised connections are used *[1 mark]*;

– (use software to) only accept text emails / ban attachments *[1 mark]* because attachments can contain viruses *[1 mark]*

– Stopping employees bring in their own discs *[1 mark]* which might include data for sending in emails *[1 mark]*. (Accept this last point, even though it is not specific to emails.)

(c) *(Award [1 mark] for a valid task, and [1 mark] for a suitable elaboration; for two tasks, giving a maximum [4 marks]:)*

– training *[1 mark]* ; an employee needs to be taught how to access/use the system to reduce errors *[1 mark]*;

– user-id needs to be added to system log *[1 mark]* so that when logging-on the new employee is recognised as an authorised user *[1 mark]*;

– level of hierarchy needs to be set *[1 mark]* so that the new user can only access data that s/he is supposed to *[1 mark]*.