1. (a) Award marks as indicated below; up to *[4 marks]* max.
[1 mark] for suitable layout, *e.g.* clear table and each point separated logically.

BOTTOM	TOP	MIDDLE	MIDDLE ITEM	SEARCH ITEM	FOUND	
1	7			Kiwi		
		4	Grapefruit		false	*[1 mark]*
5						
		6	Orange		false	*[1 mark]*
	5					
		5	Kiwi		true	*[1 mark]*

(Do **not** deduct marks for allocating BOTTOM or TOP to be MIDDLE rather than MIDDLE+1 and MIDDLE-1 respectively. Follow logic through.)

Question 1 continued

(b)   Award marks for the algorithm as indicated below; up to *[16 marks]* max.
Example is below:

```
BINSEARCH
    /* MIDDLE, TOP, BOTTOM and FRUIT are as declared in question. */
    repeat
        MIDDLE ← (BOTTOM + TOP) div 2
        if FRUIT[MIDDLE] = INPUT_ITEM then
            output ("Found")
        else
            if FRUIT[MIDDLE] > INPUT_ITEM then
                top ← MIDDLE-1
            else
                BOTTOM ← MIDDLE+1
            endif
      endif
    until FRUIT[MIDDLE] = INPUT_ITEM or TOP < BOTTOM
    if TOP < BOTTOM then
        output ("Error")
    endif
endBINSEARCH
```

–   *[4 marks]* for outer loop:
–   *[1 mark]* for **any** loop (even if incorrect);
–   *[1 mark]* for a good attempt at terminating loop if INPUT_ITEM is found (*i.e.* FRUIT[MIDDLE] = INPUT_ITEM. Alternatively a Boolean could be used and set within the procedure);
–   *[2 marks]* for correctly stopping loop if item is not found (*i.e.* TOP < BOTTOM). (*[1 mark]* for an attempt, but incorrect.);
–   *[2 marks]* for calculating the middle location:
e.g. MIDDLE ← (BOTTOM + TOP) div 2 (if <u>integer</u> division is not made clear, *e.g.* by div or truncate((BOTTOM+TOP)/2) or round((BOTTOM+TOP)/2) or int((BOTTOM+TOP)/2) (do not penalise even though int is not in PURE) *etc.*, only award *[1 mark]*);
–   *[1 mark]* for correct if to test if item is found;
–   *[1 mark]* for output at correct location;
–   *[5 marks]* for reallocating TOP or BOTTOM:
–   *[1 mark]* for a suitable if test;
–   *[2 marks]* for readjusting TOP and BOTTOM in correct location as a result of the if. (*[1 mark]* if readjusted TOP and BOTTOM, but in wrong location);
–   *[2 marks]* for setting TOP and BOTTOM to MIDDLE-1 and MIDDLE+1 respectively (*[1 mark]* for a minor error, *e.g.* assigning MIDDLE+1 to TOP and MIDDLE-1 to BOTTOM, or setting both to MIDDLE *etc.*);
–   *[3 marks]* for testing for an error:
–   *[2 marks]* for correct if test (*[1 mark]* for an incorrect attempt, e.g. if ERROR then *etc.*);
–   *[1 mark]* for a suitable error output.

Question 1 continued

(c)   Award marks for the algorithm as indicated below; up to *[10 marks]* max.
       Example is below:

```
procedure BSEARCH (ref (ARRAY IS FRUITARRAY,
                   val START is integer,
                   val FINISH is integer)
   if FINISH < START then
      output ("Error")
   else
   MIDDLE ← (START+FINISH) div 2
      if ARRAY[MIDDLE] = INPUT_ITEM then
         output ("Found")
      else
         if ARRAY[MIDDLE] > INPUT_ITEM then
            BSEARCH(FRUIT, START, MIDDLE-1)
         else
            BSEARCH(FRUIT, MIDDLE+1, FINISH)
         endif
      endif
   endif
endBSEARCH
```

Award marks as follows:

–   *[5 marks]* for correct terminating condition:
       –   *[1 mark]* for correct position, *i.e.* at start;
       –   *[2 marks]* for correct `if` (1 mark for any attempt at an `if` to terminate
              recursive routine);
       –   *[1 mark]* for error `output` in correct place;
       –   *[1 mark]* for rest of algorithm in the `else` section;
–   *[1 mark]* for an <u>attempt</u> at an `if` statement that leads to at least one recursive call
       to BSEARCH (do not penalise if the `if` statement is wrong, since this will have been
       taken into account in the previous section);
–   *[4 marks]* for recursive calls:
       –   *[1 mark]* for <u>any</u> recursive call to BSEARCH, even if very wrong;
       –   *[1 mark]* for <u>two</u> recursive calls;
       –   *[2 marks]* for all parameters correct (*[1 mark]* for a minor error, *e.g.* in
              wrong order within call).

Notes:
–   Do not deduct marks for using actual parameter names as formal parameters, *i.e.*
       accept FRUIT, TOP *etc.*;
–   Marks are not awarded for calculating MIDDLE, since this was done in previous
       section.

2.   (a)   Award *[1 mark]* for:
WAN (Wide Area Network).

(b)   (i)   Award *[1 mark]* for any suitable method, up to *[2 marks]*. For example:
–   questionnaires;
–   observation;
–   inspecting (current) paperwork, forms *etc.*

(ii)   Award *[1 mark]* for any advantage for each method, up to *[3 marks]*. For example:

interview	–	good for small sample;
	–	allows analyst to ask questions in response to an answer from interviewee;
	–	*etc.*
questionnaires	–	good at getting information from a very large sample of people;
	–	standard format of questionnaire is easier to analyse;
	–	*etc.*
observation	–	good at seeing precisely how tasks are carried out;
	–	first hand view (people are not always willing to say the truth, or they sometimes make things sound better or worse than reality);
paperwork	–	looking at forms (reports, invoices *etc.*) will help the systems analyst formulate record definitions for the various records to be kept on the system.

(c)   *[2 marks]* for a clear outline of a task; *[1 mark]* for a partial outline; max. *[6 marks]*. Examples include:
–   the system analyst carries out a feasibility study that will consider the practicality and financial aspects of installing a new system in place of the current system. (Advantages and disadvantages of the new system are pointed out within the feasibility study.);
–   documents and analyses current system operations to see where computers can be used/improvements made;
–   defines user requirements for improving or replacing systems;
–   identifies hardware and software needed in suggested system(s);
–   designs system inputs, outputs, procedures and flow of information;
–   trains users to work with new system;
–   *etc.*

(d)   *[2 marks]* for a clear outline, *[1 mark]* for a partial outline.
According to the case study, records are exchanged via e-mail. The doctor wanting to receive the record will e-mail his/her request to the hospital holding the record in question. The data administrator (or other authorised person) then sends a return e-mail to the doctor with the patient record stored in an attached file.

Question 2 continued

(e) Award *[1 mark]* for each correct measure, up to a max. of *[2 marks]*.
Examples:
- use of passwords to access records;
- encryption of data records;
- use of software to prevent computer viruses from modifying records;
- physically securing the computer room and all other rooms in hospital where data is kept against fire, flood, theft, *etc*;
- security checks of personnel;
- *etc.*

(f) (i) Award *[2 marks]* for a full answer, *[1 mark]* for a partial/vague answer. For example:
Each hospital developed its own software over the years before they worked together, and they may have used different codes. Since programmers are now creating new software to co-ordinate stock levels they may choose ANY code.
OR
As the three hospitals were each maintaining their own systems previously, it is highly likely that they would be using different coding methods for some (if not all) drugs kept in their inventory.

(ii) Award *[2 marks]* for a full description, *[1 mark]* for stating a problem without expanding. Examples are:
- The employees at the three hospitals, being used to their old systems might find themselves keying in wrong numbers in the early stages of using the new system. If some of the codes are similar, this could lead to mistakes in the allocation/prescription of drugs;
- A separate file which relates the old code(s) and the new unified code is needed if the old codes are to continue;
- Employees need to learn the new drug codes to be able to access and update the fields for its own stock numbers, which may take time and/or result in errors.

(g) *[4 marks]* for a thorough discussion with supporting arguments from the case study;
[3 marks] for some discussion with supporting arguments from the case study;
[2 marks] for some discussion but with no reference to the case study;
[1 mark] for relevant point dealt with in a trivial manner.
Examples:
If the new system allows for the portable machines that aid in diagnosis, it could turn out that less people will be admitted for examinations and testing. Demand for beds could be less as a result and the extra 15 beds would not be as big an issue. Likewise, if microsurgery also becomes a prominent part of the new system, patients will need a shorter recovery time. This too will put less stress on the number of beds needed. It is also possible that, if local doctors have access to the system, some health problems which might have led to hospitalisation could now be handled locally.
OR
If an analysis concludes that no more patients have been treated (for example with the microsurgery), or that more mistakes have been made (for example with the drug stocks), over a certain period of time with the new system in comparison to the old, then it can be said that the money will not have been well spent.

Question 2 continued

(h)　Award *[2 marks]* for a full answer; *[1 mark]* for a simple answer, *i.e.* "drug delivery".
Examples:
- As the hospitals now share a drug database showing the stock levels at the three hospitals, it will be necessary for a van to deliver drugs from one hospital to another as needed;
- Because physical objects, *e.g.* drugs, machinery *etc.* cannot be sent by electronic means.

(i)　Award marks as follows, up to *[4 marks]*.

H	R	S
0	0	1
0	1	1
1	0	1
1	1	0

Award *[3 marks]* for the truth table:
- *[1 mark]* for a correct set of four inputs;
- *[2 marks]* for correct S (*[1 mark]* for a minor error, *e.g.* giving **XOR** output).

$S = \overline{H \cdot R}$
$S = H$ **NAND** R (*i.e.* a **NAND** gate)

(Award *[1 mark]* for correctly identifying the truth table's logic gate (allow follow through, *e.g.* if truth table was wrong as for **XOR**, and candidate stated **XOR** for this part, then give the mark. However, if candidate gave **XOR** truth table, then stated **NAND**, s/he will <u>not</u> get the mark.)

3. (a)   Award marks as indicated below; up to *[7 marks]* max.

    I   Award *[2 marks]* for the correct answer; *[1 mark]* for minor error.
AB+C*DE–+

        Award *[1 mark]* for stating that this is post-order (Reverse Polish Notation).

    II   Award *[2 marks]* for correct answer; *[1 mark]* for minor error.
+*+ABC–DE

    III   Award *[2 marks]* for correct answer; *[1 mark]* for minor error.
A+B*C+D–E

    (b)   (i)   Award up to *[2 marks]* for a full explanation; up to *[2 marks]* max.
Award *[1 mark]* for a partially correct answer.

This is a dynamic data structure because:
"a new memory location can be allocated as the target of a pointer variable to a structure of the type NODE.*"*
(Do **not** accept an answer along the lines of "it uses pointers".)

        (ii)   Award marks for the algorithm as indicated below; up to *[6 marks]* max.
An example is:

```
procedure TRAVERSE (ref ROOT pointer → NODE)
    if ROOT → LEFT # nil then
        TRAVERSE (ROOT → LEFT)
    endif
    if ROOT → RIGHT # nil then
    TRAVERSE (ROOT → RIGHT)
    endif
    output ROOT → VALUE
endprocedure TRAVERSE
```

Award marks as follows:
–   *[1 mark]* for specifying a parameter that is, or nearly correct;
–   *[3 marks]* for first if:
    –   *[1 mark]* for testing LEFT before any other test;
    –   *[1 mark]* for testing with nil;
    –   *[1 mark]* for a good attempt at a recursive call with a LEFT as a parameter calling to TRAVERSE;
–   *[1 mark]* for attempting (reasonable) to call TRAVERSE with RIGHT;
–   *[1 mark]* for **output** in correct location (accept **output** VALUE).

4. (a) For either organisation, award *[1 mark]* for each of the listed steps, up to *[5 marks]* maximum:

Partially-Indexed:

- input key of record to find;
- search index sequentially;
- stop search when index entry ≥ search key;
- use address given in index;
- to load data block;
- search data block sequentially;
- stop search when record found.

Fully-Indexed:

- input key of record to find;
- search index sequentially;
- stop search when index entry = search key;
- use address given in index;
- to load data block;
- search data block sequentially;
- stop search when record found.

(b) For each organisation, give *[3 marks]* for a clear description, up to *[5 marks]* total. (Award *[2 marks]* for a good description, but incomplete, and *[1 mark]* for an answer with some credit.)

To display a file in key value order:

Partially-Indexed:

- Read the first record from the index;
- use the given address to load the data block;
- display all the records in the block in the order stored;
- (if the data file is stored in contiguous – (if the data is not stored in blocks) read next data block and contiguous blocks) continue to display all records; continue until end read records from the index of file. sequentially and continue steps until end of index.

Fully-Indexed:

- Read records from the index sequentially;
- for each record use the given address to load the data block;
- search the block for the record with the key in the index record;
- display the record;
- repeat for all records in the index.

Question 4 continued

(c)   Award *[1 mark]* for the idea that the index is smaller, so the search time is less, and the 2nd mark for the idea less disc access, so speed is better in a smaller file. For example:

When accessing files, disc access is the slowest part, so the smaller index for the partially-indexed file will require less disc reads, and so the speed will be faster.

(d)   Award *[2 marks]* for a good explanation (*[1 mark]* for a partial/vague answer) and the 3rd mark for relating it to the case study. For example:

–   For reference purposes, it may be useful to print out/display the key fields of all the drugs, which is very easy with a fully-indexed file, since only the index needs to be accessed without reference to the data records.

(If more of the details need to be displayed, then partially-indexed would be better!)

–   (Although the following is unlikely – allow it.)

If a file is very dynamic with many record additions (and deletions) during daily processing, then it is faster to use a fully-indexed file since there is less manipulation of data records (because sequential order does not have to be maintained as in partially-indexed files). Since many new drugs are being made, this would make their addition to the file easier.

5.   (a)   *[1 mark]* for each function stated, up to a maximum of *[5 marks]*.

128 MB RAM
The random access memory is the volatile internal memory of the computer where programs and data can be read and written.

8 GB Hard disk
A hard disc drive is the external permanent storage medium.

256 kB Cache memory
Cache memory is a volatile, read/write memory directly associated with the processor. It acts as a go between for the processor and RAM.

400 MHz Processor
The processor is where instructions are decoded and executed.

2 Serial & 2 Parallel Ports
These are access points to which peripherals can be connected.

(b)   *[2 marks]* for each good explanation, *[1 mark]* for each partial explanation, max. *[10 marks]*.

128 MB RAM
128 megabytes of RAM is large and it means either that large programs can now be run which could not on a machine with smaller RAM or program execution is made faster since less disc access/overlaying is needed.
Larger programs are stored during execution here.

8 GB Hard disk
8 gigabytes is extensive and thus more programs and/or data can be stored than on a machine with a smaller disc capacity. A large capacity disc drive provides ease of storage as opposed to many floppy discs.
All a user software *e.g.* word processor or games, can be permanently installed here.

256 kB Cache Memory
Cache provides faster access than standard RAM and therefore program execution is made faster. Commonly used instructions from a program are stored here and the processor is able to more quickly find them.

400 MHz Processor
Programs are executed faster than in machines with a slower processor (all other factors being equal). The decoding and execution of user programs will be noticeably faster than on computers with a slower processor. Multitasking will be dealt with more efficiently.

2 Serial & 2 Parallel Ports
With more peripherals available on the market, older machines may not have enough external links. Having both serial and parallel ports allows for devices with different interfaces to be connected to the computer.