THE BCS PROFESSIONAL EXAMINATION
Certificate

October 2003

EXAMINERS' REPORT

Software Development

General Comments on Examination Technique

Candidates appear to give too much reliance on memorised code and do not spend enough practice in developing code systematically. Nearly all candidates showed lack of ability in algorithmic development, while memorised code was frequently used, sometimes in unexpected places. Thus code for sort routines was provided in questions that had nothing to do with sorting.

It is important that candidates, in preparing for the examination, tackle several problems where it is obvious to that algorithmic development is vital before beginning to write code. Other than for trivial examples, it is not possible to go from a problem to code - design is required and tested.

Candidates frequently ignored instructions not to re-write parts of an answer in different parts of a question. This is never needed - where appropriate an indication should be given where an earlier part belongs. Such copying is not penalised by the examiner but time is wasted in copying out something already written earlier which is in effect a penalty. Repeated parts never gain any marks, particularly when the question instructs the candidates not to do this.

It is important that candidates READ the questions carefully and answer what is asked.

Question 1

The following table contains data for SLR lenses available from a photographic dealer:

Focal length/mm	28	35	50	300
Wait time/weeks	0	3	0	6
Price/£ sterling	120	165	67	270
Weight/kg	0.25	0.26	0.31	0.64
Supplier/10 characters	Supplier-1	Supplier-2	Supplier-1	Supplier-3

a) Define a data structure suitable for holding the data in one column of the above table. Define a table/array capable of holding up to 50 such entries. State the language you have used. **(5 marks)**

b) Write the code to initialise the table/array defined in part *a).* The data values will be preceded by an integer value which defines the number of table/array entries to be read in. (**7 marks)**

c) Using this table/array, develop an algorithm for the following interactive process:

> A customer is asked how many lenses with different focal lengths they wish to enquire about. For each focal length the customer is invited to input a specific focal length. The table/array is then searched for this focal length. If it is present the algorithm displays *"lens is available"* and if the wait time is zero it displays *"from stock"*, otherwise it displays the wait time from the table followed by *"weeks"*. It then invites the customer to order the lens. If the reply is "**Y**" the price is added to the total. If the entry is not present in the table, it displays *"not available – sorry"*.
>
> When all the enquiries have been processed, the algorithm displays the total cost of the ordered lenses.

Develop this algorithm to a stage where translation to the target language is straightforward. **(18 marks)**

The answers given in parts *a)* and *b)* should not be repeated in *c);* just indicate where they would be placed in the algorithm.

Popular; full range of marks awarded. Generally well done by those adequately prepared in algorithmic development. Obviously many candidates are now adequately prepared for this type of question. This is better than in previous years although there is still room for improvement.

Answer Pointers

a) Usually correct when answered by those who knew Pascal or C.

b) Less well done, particularly by those who used a database language - they had no idea how to populate the tables! A very common mistake was to expect all 50 data items input when the question stated quite clearly that the actual number was to be requested interactively before any data items were input.

c) Many styles were displayed here. Some candidates had many small procedures containing (at most) one or two actual instructions. This is inappropriate if the procedures are only called once! Development was usually at too low a level, not much removed from code, consequently omissions were not detected, important instructions were in the wrong places, notably were the main loop was concerned.

It was FOOLISH to attempt this question if one had NO idea of how to answer part (c); over half of the marks were automatically lost.

Repetition of variable declarations were made in many answers despite the question stating that this was NOT necessary.

Question 2

Part of a structured text analysis program is given below in pseudocode:

```
PROCEDURE Findword has one parameter which is the input file name
    Assumption – always at least one character to be read
BEGIN
    Define a local character variable – ch
    REPEAT
        Read the next character from the input file into ch
    UNTIL ch is a letter
END

PROCEDURE Skipword has three parameters; one is the input file name, the second is an integer
    parameter (letter_ct) capable of returning a value and the third (last_ch) is a character parameter
    capable of returning a value;
    Assumption – always at least one character to be read
    Assumption – input is terminated by a standard punctuation mark
BEGIN
    Define a local character variable – ch
    REPEAT
        Read the next character from the input file into ch
        ADD 1 to the parameter letter_ct
    UNTIL ch is NOT a letter
    Set last_ch to ch
END

{top level – execution begins here}
    BEGIN
        ASSIGN the input file
    Declare integer variables letter_ct and word_ct and set them to ZERO
    Declare a character variable last_ch and set it to a space character
    REPEAT
        Call Findword with the input file as the parameter
        Call Skipword with the input file, letter_ct and last_ch as parameters
        ADD 1 to word_ct
    UNTIL a punctuation mark is reached
    Output the average word length; i.e. letter_ct/word_ct
    CLOSE the input file
END
```

a) Dry run this code with the characters **ab□c.** from the input file: [□ represents a "space"]. **(18 marks)**

b) State what the expected result would be and why the code does not reproduce this result. **(8 marks)**

c) Show how to implement the conditions "*ch* is a letter" and "a punctuation mark is reached" in a procedural language. State which language you are using. **(4 marks)**

By contrast, this was very unpopular, with most answers having an unsystematic approach and with much carelessness of detail. Some had a complete misunderstanding of the question, and re-cast the given code in Pascal (or C) without any procedures. This was DEFINITELY NOT what was asked!

Candidates should NOT have attempted this question whose preparation obviously did not include how to do dry runs with a study of previous questions similar to this. In particular, those who studied JAVA or SQL alone were not appropriately prepared.

Answer Pointers

Rough work was especially unnecessary in this question, as the point of the question was to illustrate a systematic execution of the given pseudocode. There were a dozen crossed-out attempts at this question, more than for any other, which must have wasted a significant amount of time. Others attempted it as well as two other 'A' section questions, when it invariably scored the lowest mark. Parts b) and c) were usually ignored, but there were a few good answers, notably by those trained to use COBOL although it is actually simplest in C or Pascal.

A table similar to that below was looked for.

Instruction	**Test**	**ch**	**last_ch**	**other_ct**	**letter_ct**	**word_ct**
Top level:Assign inputfile	**?**	**?**	**?**	**?**	**?**	**?**
Set counters to 0	**?**	**?**	**?**	0	0	0
Set last_ch to space character	**?**	**?**	**' '**	0	0	0
REPEAT		?				
call Findword		?				
Findword: REPEAT		?				
READ ()		a				
UNTIL ch is a letter	True					
. . . . .						

Turn over]

Question 3

a) Write a function that implements a hashing algorithm to perform a table-lookup of the following data without searching. The function should return an integer value that is the array position of the given value calculated by the hashing algorithm – in this problem the array position is in the range 1 to 5.

For example, if the function is called HASH, then HASH (6945) would return the value 5.

4213
5321
9526
1778
1634
6945

(15 marks)

b) Describe how the collisions happen when using hash-coded table lookup. Give an example to illustrate your answer. **(9 marks)**

c) Describe how collision avoidance reduces the efficiency of a hash-coded table look-up. **(6 marks)**

Most candidates who attempted this question recognised what was required and answered the question reasonably well.

Answer Pointers

a) The candidates were expected to experiment with the data to find how it can produce the index. One method is to add each digit and truncate to least significant digit.

b) Collisions happen when two different data values hash to the same position. E.g. HASH(2253) will produce same result as HASH(5322)

c) Efficiency of table lookup is O(1) because no scanning, direct access. Avoidance usually involves noticing the collision then scanning for the alternative, correct place by content-matching rather than calculation. The scanning reduces efficiency, on average, to O(n/2).

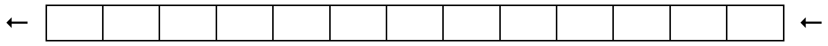
Question 4

a) Describe the operation of a queue, including how elements are inserted and removed. Be sure to include the special cases of inserting in to a full queue and removing from an empty queue. **(15 marks)**

b) Describe the operation of any piece of system software that might use a queue data structure as part of its data store. Illustrate your answer with a description of what the system software does and draw suitable diagrams. **(15 marks)**

Answer Pointers

a) A queue is FIFO First In First Out. Code that implements a queue knows about the head position, the tail position, the current number in the queue and the max size of the queue. Head and Tail are operated with modulo-MAX arithmetic to achieve wrap-around.



FIFO

Initially, HEAD = TAIL. Number-in-Queue=0, MAX is some arbitrary capacity;

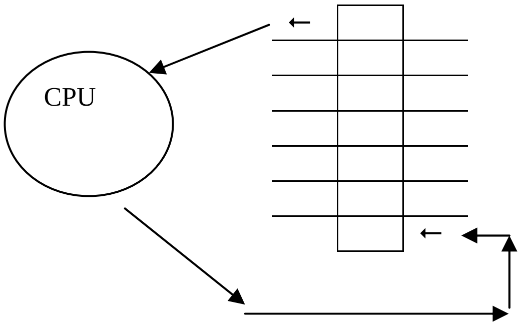
INSERT: if Number-in-Queue=MAX then return error “overflow –queue full”
else insert at TAIL
(Increment TAIL) mod MAX so that TAIL wraps-round. *[7 marks]*

DELETE; if Number-in-Queue=0 then return error “underflow –queue empty”
else delete from HEAD
(increment HEAD) mod MAX so that HEAD wraps around. Note increment.

An alternative answer might involve the same set of operations on a linked list.

b) A suitable piece of system software is the CPU scheduler with a round-robin despatch algorithm. Tasks are queued for access to CPU. A task is stopped after max-time (so-called CPU-intensive) or when it makes an i/o request within the period max-time. Task is then queued to wait for next opportunity for access to CPU.

A suitable diagram might look like



Question 5

A rubber ball is dropped on to a hard flat surface from a height “H”. Each bounce returns it to height “P” percent of the previous bounce height where P is less than 100%. Eventually the height of the ball reached is less than an input minimum, “U”.

Develop the logic and write a short program to implement this problem and count the number of bounces made before the height is less than “U”.

Incorporate a feature which stops the program running should the bounce count exceed 10000. **(12 marks)**

Quite popular, full range of marks awarded. Higher marks were given to those who tried to develop the logic, (even if errors were made) than those who plunged straight into code. Again candidates were careless of detail (or only read the question once) and so forgot to input ALL the necessary data items needed, or output the result.

Answer Pointers

If bounce height is ‘B’ then B = P(percent) * H(height) /100
If the minimum is not reached then H becomes B and repeat the process, incrementing a counter (ct)
We also need a loop terminating condition if ct exceeds 10,000.

```
INPUT   initial height H
INPUT  minimum bounce height U
INPUT  percentage return P
Set ct to 0
WHILE H >= U AND ct < 10000 DO
        B = P * H / 100
        Increment ct
```

```
    H = B
ENDWHILE
```

Question 6

Accurate values of the cosine function(X) are calculated from the infinite series thus:

$$Cos(X) = 1 – X^2 / fact(2) + X^4 / fact(4) + X^6 / fact(6) – X^8 / fact(8) + …$$

where fact(N) = N*(N-1)*(N-2)*…*2   i.e. factorial (N).

a) Write a FUNCTION to calculate fact(N) where N is a REAL number. Any method may be used. **(3 marks)**

b) Show how each term in the series may be derived from its predecessor by an appropriate multiplier. Hence show that a separate factorial function is not needed to evaluate the series. **(3 marks)**

c) Write an algorithm OR function to obtain cos(X) to a precision of 10 decimal places. **(6 marks)**

An unpopular question, as is anything with a mathematical content. It was not sensible to attempt this question if all one knew was memorised code for *factorial(n),* restricted to 3 marks. Very few were able to derive the relationship between successive terms, but answers implementing each term calculated separately were allowed as correct, particularly if they showed where to use the factorial(n) code written earlier.

Answers which implemented the ellipsis (…) in their code were penalised, as this suggests a complete misunderstanding of how to tackle the question. Besides, no procedural languages allow terms to be implied in this way.

Answer Pointers

a)

```
FUNCTION  FACT( N : REAL ) REAL;

BEGIN
        IF N = 1.0 THEN fact := 1.0
                  ELSE fact := fact (N – 1.0) * N
END;
```

b) $\cos(X) = \underbrace{1}_{\text{term }(0)} - \underbrace{X^2 / fact(2)}_{\text{term}(1)} + \underbrace{X^4 / fact(4)}_{\text{term}(2)} - \underbrace{X^6 / fact(6)}_{\text{term}(3)}$

after term(0) each term (N) may be obtained by multiplying its predecessor by

$(-1)*X^2 / [N*(N-1)]$

Thus $X^4/ fact(4) = X^4 / 4.3.2.1 = x^2 / 2.1 * (-1)* x^2 / 4.3$

$X^6 / fact(6) = X^6 / 6.5.4.3.2.1 = x^4 / 4.3.2.1 * (-1) * x^2 / 6.5$

c) This part of the question just required the absolute difference between the previous value and the current value of cos (x) to be compared and to check whether it was less than the required difference.

Question 7

A serial file ‘datafile’ has a sequence of records R_1, R_2, …, R_N. A file pointer is used to manage operations with this file.

a) Draw a diagram showing how the records are laid out in the file. Include the file pointer’s position before any records are read from the file. **(4 marks)**

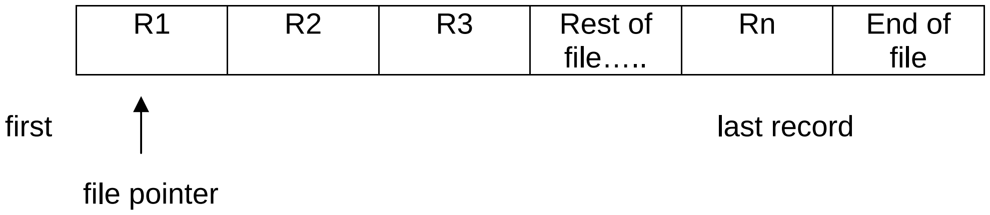
b) Show diagrammatically or describe how the END-OF-FILE (datafile) condition is detected. **(4 marks)**

c) Write a program loop which opens the file, counts how many records are in the file and then closes it. State which language you have used. **(4 marks)**

Surprisingly unpopular as descriptive questions usually attract many answers. Files were frequently confused with linked lists even to writing linked list code. Others wrote lots of irrelevant material on files, which wasted time. Candidates should read the question and decide just what is wanted, rather than latch onto a keyword like ‘files’ or ‘pointers’ and then perform a ‘brain dump’ of everything they know on the subject. Again this was (usually) not penalised, although some wrote two sides on what actually took only 2 - 3 sentences and an annotated diagram. In particular, part (a) did ask specifically for a diagram, so those who did not give one were penalised.

Answer Pointers

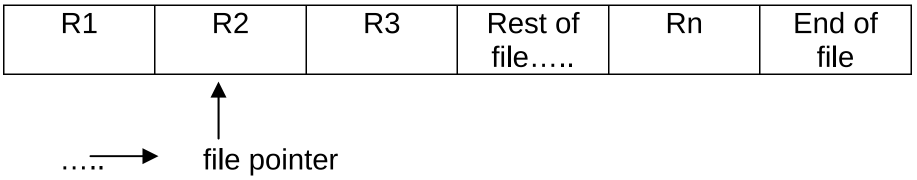
a)



R1	R2	R3	Rest of file…..	Rn	End of file

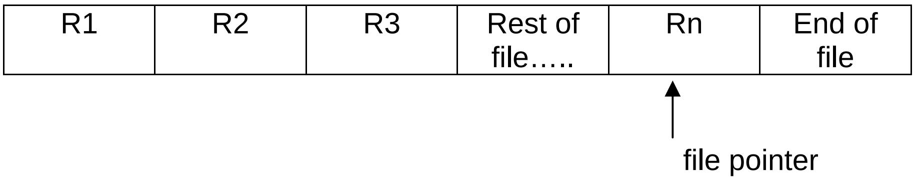
file and associated pointer after executing the ‘reset’ instruction. Note that the file pointer indicates the start of the NEXT record on the file that is available.

After executing one READ (datafile) instruction the file pointer moves down the file by one record.



R1	R2	R3	Rest of file…..	Rn	End of file

End-of-file becomes true when the LAST record has been read. On most systems the actual end-of-file marker is not read over, although it used to be in COBOL programs. An error condition results if another READ operation is now attempted.



R1	R2	R3	Rest of file…..	Rn	End of file

c)

```
OPEN datafile as #1
recordct = 0
WHILE NOT EOF #1 DO
        READ (datafile, onerec)
        increment recordct
ENDWHILE
CLOSE #1
```

Question 8

a) Define a suitable data structure to hold a node of a linked list with one pointer and 3 data items, these being the title of a book (up to 20 characters), the price (dollars and cents) and the number of books sold. **(3 marks)**

b) Write a PROCEDURE ‘onend’ to add a new node at the end of the list. **(5 marks)**

c) Input the values for a new node interactively with suitable prompts and invoke your procedure 'onend' with appropriate parameters. Assume that the list is set up beforehand by PROCEDURE ‘invoke’ which you are not expected to write; show where it is invoked in your code. **(4 marks)**

This distinct part of the syllabus is often left out, so it attracted few answers. Generally the few who did this question scored well, except for those who answered only the declaration, part (a), and restricted themselves to 3 marks. Those who did list traversal to find the end were given credit if it was in procedure *onend()* (as intended) or in the top level expected in (c). Quite a lot did not realise it was needed, of course.

Answer Pointers

a) One possible data structure would be

```
TYPE  aptr = ^ node;
node = RECORD;
        title : PACKED ARRAY [1..20] OF CHAR;
        price : REAL;
        numsold : INTEGER;
        next : aptr
               END;
```

Parts b) and c) where looking for standard code to add new nodes.

Question 9

a) Describe the form of file organisation that you would use for a file of invoices that need to be printed for posting to relevant customers. Be sure to include in your answer the reasons why this form of file organisation is suitable for this application. **(6 marks)**

b) Describe the form of file organisation that you would use for a file of accounts that need to be queried interactively as customers telephone in to ask about their account status. Be sure to include in your answer the reasons why this form of organisation is suitable for this application. **(6 marks)**

Answer Pointers

a) Either SEQUENTIAL or INDEX_SEQUENTIAL will support this operation because it is essentially a batch operation to pass over the whole file in O(n) time and print details from every record.

The SEQUENTIAL organisation is good at quickly fetching sets of records in sequence

b) Either INDEX or INDEX-SEQUENTIAL will support this operation because it is a quick, direct lookup operation O(1) to retrieve a specific record from the whole set.

The INDEX organisation supports direct access by maintaining an index to each record, for speed of retrieval.

Question 10

You are asked to design the Concepts and Facilities Guide for a small-sized database contact-management application. Describe the sections of the Guide that you think are important and useful to the User. Give your reasons. **(12 marks)**

Answer Pointers

“Concepts” will be the essential features of a contact-management application. Typical features are

record details of a meeting/phone call/client
retrieve details of a meeting/phone call/client
summary report number of contacts in unit time
summary report specific demographics of contact (order size, location, etc.)

Essentially, these are cognitive-specific features about a contact management system.

“Facilities” will be the standard set of actions on a database application – how to create, read, update, delete; plus descriptions of permissions (who-can) at each level of access

Question 11

Describe THREE GUI features that you expect to find in the design of an interactive website for an e-commerce company that sells books. Give your reasons. **(12 marks)**

Answer Pointers

What is wanted is some discrimination that picks up keywords ‘interactive website’ and ‘e-commerce company that sells books’ to decide the user is a B2C client who wants to pick a book. Keeping client ‘in control’ of the dialog is the aim of the GUI, achieved by such things as

drop-down lists - for specification of book reviews, or price variation.
Search engine – to produce lists of possibles and alternatives.
Frames approach - to maintain navigation while offering details of books, similar books and prices
Radio buttons – for selection of preferences such as payment option, postage class.

Question 12

Briefly describe the roles of:

a) Database Administrator (DBA) **(4 marks)**
b) Network Administrator **(4 marks)**
c) Systems Programmer **(4 marks)**

Answer Pointers

a) A Database Administrator is responsible for the installation, configuration, upgrade, administration, monitoring and maintenance of database(s) and associated software products, in support of operational systems and the development environment. An alternative role-definition might follow the Create/Read/Update/Delete entity lifecycle.

b) The Network Administrator is responsible for day-to-day network support, including resolution of network problems, data backup and restore, production of network performance statistics, and provision of network diagnostic information; i.e. user problem resolution, network performance measurement, data security/safety, and network service maintenance.

c) The Systems Analyst is responsible for the methodical investigation, analysis and documentation of all or part of a business in terms of business functions and processes, and the information they use. Also, for definition of requirements for improving any aspect of existing processes and systems. A good answer will locate the analyst mostly on the application area (business, usually) and make them responsible for documenting the specification with process descriptions and data definitions