THE BCS PROFESSIONAL EXAMINATION
Certificate

April 2003

EXAMINERS' REPORT

Software Development

General Comments

Once again a number of candidates wrote out the answer in rough form and then wrote it out again neatly writing the answer to a question then writing it again gains no extra marks only wastes time. Lengthy crossings-out again wastes time. Candidates must learn how to write an acceptable answer without having to write it out **fully** in rough first – there simply is not enough time. It is often useful to jot down notes/key points as an aide mémoire prior to writing out the full answer but there is not time to write the answer in rough and then in a neater hand.

Writing out the question prior to answering wastes time and never gains any marks.

Question 1

1. Write a program that illustrates the growth of the world's population using as input:

the start year
the initial population size
the annual percentage increase of the population
the end year.

You may assume that the input is valid.

Output the year, the size of the population and the population density every decade:

Surface area of a sphere is $4*\pi* R^2$
Earth's radius [R] =3,984 miles
Fraction of surface which is land [F] = 0.29
π = PI = 3.14159

Use the total surface area of the earth (from given values for the Earth's radius [R]) and the fraction of the earth's surface which is land [F] to compute the land area. Specify the language used.

Initial Algorithm

CONSTANTS earth's radius, PI, fraction of surface which is land
DECLARE necessary variables
CALCULATE the land area
INPUT the start year and the start population size
INPUT the annual percentage increase of the population
INPUT the end year
CALCULATE the annual growth rate
CALCULATE the number of decades (rounded up to the next highest integer)
CALCULATE initial population density {the initial population / land area}
PRINT headings
PRINT start year, initial population, initial population density
SET working population to the initial population size

FOR count = 1 TO number of decades DO
    FOR inner count = 1 to 10 DO
        CALCULATE the next working population {working population * growth rate}

END FOR
CALCULATE the population density
UPDATE current year
PRINT current year, the working population and the population density
END FOR **(30 marks)**

This was often a first-choice question but too many candidates wasted valuable time by one or even two full pages of crossing-out. Candidates MUST learn how to do this sort of question without lots of rough work.

Answer Pointers

Little algorithmic development was needed so more importance was attached to correct language constructions. In particular, marks were given for:-

i) Meaningful names for variables,
ii) Captions and formatted output for results,
iii) Appropriate types for variables,
iv) Correct scope of the main loop,
v) Appropriate use of functions/procedures.

Question 2

2. *integer function* **Afind (***input integer* **key)** is a lookup search function that has access to an external function named **calc**. The **calc** function takes an integer and returns an integer value in the range of a **dataset** as a look-up value.

```
line    code
1       Afind(input integer key);
2       begin
3           local integer pos
4           pos := calc (key)
5           if key = dataset (pos)
6           then Afind := pos
7           else Afind := 0
8       end
```

a) Suppose you have the following **dataset (a)** that requires an index in the range 1 to 6. For each value in **dataset (a)**, the function **calc** should return its location. For example, **calc (9167)** should return the value 3. Write an algorithm for the function **calc,** suitable for **dataset (a),** using pseudocode or structured English.

1	4511
2	8725
3	9167
4	1445
5	1527
6	3995

dataset (a)

b) The function **calc**, when applied to the values in **dataset (b),** would return the values 1, 2, 3, 4, 2, 5 respectively. That is, the values 1001 and 3200 are out of position according to the value returned by **calc**. How would you modify **Afind** so that out of position values are located correctly? You may use pseudocode or structured English to describe your changes. (Do NOT modify the function **calc**.)

1	5411
2	8752
3	1967
4	4145
5	1001
6	3200

dataset (b)

Very few tackled this question, and even fewer understood what was wanted.

Answer Pointers

A number of algorithms are possible. The examiners were looking for clarity of thought.

Function calc - one parameter (an integer) and returns a value in the range 1 to 6
{add the digits of the number together and use the MOD operator}

Initialise the variable "sum"
Add the digits of the parameter (via a loop using the DIV and MOD operators) into "sum"
Apply MOD 7 to the "sum" - results in a value 0 to 6
If the result is zero, add one to the result - not strictly necessary for the values in "dataset" but if applied to a different data set would require this test.
Return the result

For part b), require Afind to loop through the data set to handle the case where to values map onto the same value. Checks are required for coming to the end of the data set (i.e. need to go to the start for the next element) and to stop once looked at each element once.

Question 3

3. *a)* Describe the functions a CASE tool should contain. **(6 marks)**

b) Describe the features and function of ONE CASE tool that you know or have used. (**12 marks)**

c) Describe TWO ways in which the use of CASE tools is changing the traditional method of software development. Give reasons for your answers. **(12 marks)**

*Some candidates confused 'benefits' with 'features'. Benefits such as speed, productivity, or standards **do not** substitute for functional capability like code generation, database filing and design drawing.*

Answer Pointers

The fist part of the question required a standard definition of an ideal CASE tool. The second part drew on the experience of the candidate's use of a specific CASE tool. For example, the methodology supported by the CASE tool and how the functions and features of the tool supported the methodology.

The third part of the question required the candidate to think about the impact of CASE tools and how they affected the development life cycle. There was no right or wrong answer to this part of the question. The examiners were looking to see how the candidate could analyse a situation and draw reasonable and sensible conclusions.

Question 4

4. The algorithm given below determines the highest common factor (**hcf**) of two given positive integers m and n.

```
r = m MOD n
WHILE (r NOT EQUAL TO 0) DO
        m = n
        n = r
        r = m MOD n
END WHILE
PRINT "highest common factor =" n
```

a) DRY RUN this algorithm using m = 242 n = 154. **(15 marks)**

b) Using a procedural language of your choice, translate the algorithm into a program which asks for m and n as input values. The program should also count and print how many iterations were required to obtain the **hcf**. Use meaningful variable names. State which language you have used.

c) Convert the algorithm to an equivalent recursive algorithm. **(5 marks)**

This question was very popular with nearly all candidates attempting it. The full range of marks was used.

The dry run itself was generally well done. Candidates will normally find that using a tabular form for the answer is both clearer and quicker to write.

Answer Pointers

a) High marks were only gained by those who demonstrated the loop and register contents for most if not all instructions. FIVE columns in the table were expected for m, n and r variables, line number/instruction and results; e.g. whether boolean expressions were true or false. Careless arithmetic errors were penalised only once. (Some still contrived to get the right answer!) Just a few did not know what 'MOD' was and carried out real division or subtraction. Unheaded columns of figures or not showing at least one full loop execution was penalised.

Instruction	Working	m	n	r
Start		242	154	?
r =m MOD n		"	"	88
WHILE ()	True	"	"	"
m = n		154	154	"
n = r		154	88	"
r = m MOD n	154 MOD 88 → 66	"	"	66
WHILE ()	True	"	"	"
m = n		88	88	66
n = r		88	66	66
r = m MOD n	88 MOD 66	"	"	22
WHILE ()	True	"	"	"
m = n		66	66	22
n = r		66	22	22
r = m MOD n	66 MOD 22 → 0	"	"	0
WHILE ()	false/loop ends			
PRINT	Highest common fact = 22	66	22	0

b) Generally well done. Marks were given particularly for 'added value' like declaring variables properly, including an iteration counter, meaningful names for variables (single-letter names rarely imply meaning) and captioned output.

c) Marks were given for a recursive call (whether or not the parameters were correct) and a recursion terminator. Even so, most candidates scored low marks for this question.

Question 5

5. The slope (m) and intercept (c) for the straight line y = m*x + c can be calculated from the following formulae, which apply the linear least squares method:

Sum_x = sum of all x_values
Sum_xy = sum of x*y values
Sum_y = sum of all y_values
Sum_xx = sum of x^2 values

m = [Sum_x * Sum_y - N * Sum_xy] / [$(Sum\_x)^2$ - N*Sum_xx]
c = [Sum_y - Sum_xy*Sum_x] / [N * Sum_xx - $(Sum\_x)^2$]

a) Develop an algorithm which inputs N pairs of (x,y) values and which calculates m and c from these formulae. **(6 marks)**

b) Write a program corresponding to your algorithm. **(6 marks)**

Few students knew what belonged in the algorithm and what was best relegated to the coded program.

Answer Pointers

The algorithm is the PLAN for the code, and should have no more detail than necessary to check that the necessary statements are in the right place. Some candidates

included detailed expressions for the slope and intercept but without any loop to input all the (x, y) pairs. It is also acceptable to state 'calculate m and c from given formulae' rather than copy out the long expressions given in the question.

The program should include 'added value' like variable declarations, prompts for input, captioned output. In particular these captions should be 'slope' and intercept' rather than 'm' and 'c' - program users would not know what the letters meant.

Question 6

6. *a)* Write a FUNCTION named 'triang' whose parameters (a,b,c) are three real numbers specifying the lengths of the sides of the triangle. The function is to return TRUE if (a, b, c) can form the sides of a closed triangle. Otherwise it returns FALSE. Use as a test that to form a closed triangle the sum of any pair of the numbers must be greater than the third number. **(4 marks)**

b) Embed this function in a test program which asks for the numbers to be input. If input values form a closed triangle, the area of the triangle is to be output using the formulae:

$$s = (a+b+c)/2$$
$$area = SQRT(s*(s-a)(s-b)(s-c))$$

If the numbers do not form a closed triangle, the message "sides do NOT form a closed triangle' is to be output instead. **(8 marks)**

Quite popular and generally done well, particularly by those candidates who attempted both the programming questions in section A. Some candidates did not realise just what the function was expected to do, despite being told in the question.

Answer Pointers

Many candidates had only one test on the triangle sides (e.g. a + b > c) when three were needed, or sorting the sides into ascending order first. Many left out the multiply (*) sign between the brackets needed in the coded expression. Others wasted time writing out the function code twice when all that was needed was a statement like 'call function triang here'.

```
FUNCTION triang(a1,b1,c1: REAL):BOOLEAN;
BEGIN
        IF (a1+b1 > c1) AND (b1+c1 > a1) AND (a1+c1 > b1) THEN
                triang := TRUE
        ELSE  triang := FALSE
END
```

Question 7

7. Specify an algorithm which reads into a variable a large integer, scans it and prints the digits it contains in ascending order. Convert your algorithm into program code. State the language used.

For example, the integer value 9089867 would produce the output 0 6 7 8 9. **(12 marks)**

Many candidates thought this just involved sorting the digits so they wrote out memorised code for a simple sort routine without any indication of how the separate digits were got from the given input single integer. It clearly showed how few students have any grasp of algorithmic development, let alone produce the individual digits of a number by 'MOD' and 'DIV' operations. Others should have realised that a 'B' section question would not expect up to FOUR pages of code involving at least 4 arrays.

Answer Pointers

Algorithm using sets

Read the value into "longval"
Set "digset" to the "empty set"
Find one digit at a time until none left (Repeat)
Extract the least significant digit - set "digit" to "longval" MOD 10
Remove the least significant digit - set "longval" to "longval" DIV 10
Add the digit just extracted to the set if not already in the set -
if "digit" NOT IN "digset" then add "digit" to "digset"
UNTIL longval is zero; that is, all digits extracted

Print the values in "digset" in order

Question 8

8. A linked list has been set up using the following node structure:

```
node_ptr  = ^node;
node   = RECORD
                a_number : INTEGER;
                next : node_ptr
         END;
```

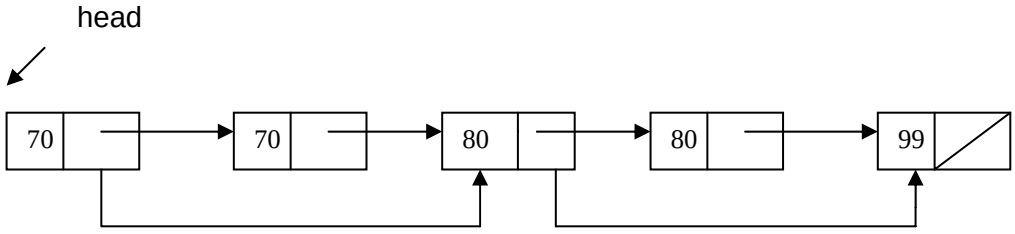
The list contains an increasing sequence of integer numbers, some of which are the same and next to each other in the linked list.

a) Show, with the aid of diagrams, the pointer movements needed to remove duplicate numbers from the list.

b) Specify the algorithm (using the given record structure) to carry out the removal of the duplicates and to report afterwards on how many removals took place. **(9 marks)**

There were some good answers. Other candidates worked from memorised code and did not check if it answered this particular problem

Answer Pointers



If consecutive data items are the same, the pointer of the first item needs to point to the next but one down the list, as shown. (i.e. if item1 = item2 then the ptr of item1 needs to be set to the ptr of item2)

Algorithm

```
Set "counter" to zero
If "head" is not nil then (that is, the list is not empty)
  begin - at least one element in the list
        Set "item1ptr" to "head"
        While not looking at the last element (i.e. the "next" field of "item1ptr" is not nil)
        begin
              Set "item2ptr" to the "next" field of "item1ptr"; that is, "item1ptr" and
              "item2ptr" are pointing to consecutive elements in the list
              if the two items are the same (if "data" field of item1ptr = "data" field of
        "item2ptr") then
                begin
                      Skip the second item (that is, set the "next" field of "item1ptr" to
                      the "next" field of "item2ptr")
                      Increment the count of items removed by 1 (that is, add 1 to
              counter)
                end
              else move to the next item (that is, "item1ptr" is set to "next" field of
"item1ptr")
        end
  end
```

Question 9

9. A database application, developed by a software house for a client, requires the client's programming team to regularly upgrade and maintain the database structures. Describe the type of documentation the software house should provide the client's programming team in order for them to undertake this task. Give reasons for your answer. **(12 marks)**

Answer Pointers

As this software has been developed by an outside agency, there is a requirement for user, system maintenance and testing documentation. All the documentation needs to be in sufficient detail so that people not involved in the original design and implementation are able to support the software.

Issues that were rewarded were

- features and facilities for teaching the client's staff how to use it,
- test cases and test history to assist maintenance programmers,
- design and code documents to assist maintenance programmers.

or a discussion that identified and distinguished between perfective and adaptive maintenance.

Question 10

10. Describe when the following types of software tests are suitable. Give reasons and examples to support your answers.

a) White box testing and Black box testing together **(6 marks)**

b) Black box testing alone without White Box testing **(6 marks)**

Answer Pointers

This question asked for a focussed answer about testing in the particular scenarios given. In part (a), most candidates recognised the 'unit level' code-&-test activity associated with white and black box testing. Candidates were expected to distinguish successfully between White Box (internal, testing all paths) and Black Box (external testing only of inputs and outputs).

Few marks were awarded if clear distinction between the two types of testing was not given. Some candidates attempted to define Black Box by writing that it was "not White Box". Such definition by negatives is generally unhelpful and not rewarded.

Question 11

11. Describe how you would construct an ***abstract data type***. Your answer should consider at least TWO examples with descriptions of how to use them. **(12 marks)**

Answer Pointers

This question sought assurance that the candidate knew about object programming, or the principles of information hiding. Candidates who described packaging of code and data behind an interface of defining functionality were rewarded.

However, candidates who then went on to exemplify 'information hiding' with two pages of detailed algorithmic description were held not to appreciate the purpose of data abstraction – to conceal implementation and reveal only functionality. Candidates who sketched only interface details of some objects that required code and data – such as stacks, queues, lists and the like – were rewarded.

Question 12

12. Describe the function of these system software processes:

a) Linking and loading **(6 marks)**

b) Compiling and running **(6 marks)**

Answer Pointers

Not many candidates understood these system software processes thoroughly. Many candidates could describe 'linking' as joining up pre-compiled modules such as library routines and deciding relative base + offset addresses. Fewer could describe loading as installing the code in memory and resolving absolute addresses according to the memory locations occupied.

Very many candidates successfully described compilation as a process of translation and calculation of relative addresses. Very few indeed understood how loading took place, that it is a two-step sequence of linking and loading as in (a), or may be packaged into a programming environment as a compile, run and test process that nonetheless requires linking and loading after successful compilation. The 'running' occurs in memory on the processor after linking and loading.