THE BRITISH COMPUTER SOCIETY

THE BCS PROFESSIONAL EXAMINATION
Certificate

SOFTWARE DEVELOPMENT

24th April 2003, 2.30 p.m.-4.30 p.m.
Time: 2 hours

SECTION A

Answer TWO questions out of FOUR. All question carry equal marks.

*The marks given in brackets are **indicative** of the weight given to each part of the question.*

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
$\pi = PI = 3.14159$

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)**

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)

(10 marks)

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)

(20 marks)

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)**

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.
                                                                                                    (10 marks)

c)   Convert the algorithm to an equivalent <u>recursive</u> algorithm.   **(5 marks)**

SECTION B

Answer FIVE questions out of EIGHT. All questions carry equal marks.

*The marks given in brackets are **indicative** of the weight given to each part of the question.*

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)**

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)**

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)**

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.
(3 marks)

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)**

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)**

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)**

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)**

12. Describe the function of these system software processes:
    a) Linking and loading **(6 marks)**

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