UNIVERSITY COLLEGE LONDON

University of London

EXAMINATION FOR INTERNAL STUDENTS

For The Following Qualifications:-

B.Sc. *M.Sci.*

Mathematics M253: Computational Methods

COURSE CODE	:	**MATHM253**
UNIT VALUE	:	**0.50**
DATE	:	**04-MAY-05**
TIME	:	**10.00**
TIME ALLOWED	:	**2 Hours**

05-C0971-3-50

TURN OVER

All questions may be attempted but only marks obtained on the best **four** *solutions will count.*
The use of an electronic calculator is **not** *permitted in this examination.*

1. (a) (i) What is the base 2 (binary) representation for 23.125?
   (ii) What is the base 2 representation of 1/7?

   (b) Express the integers -34 and 59 in 8-bit binary according to the 2's complement method. Show how the two binary numbers may be added to find the 2's complement representation of 25.

   (c) Find five errors in the following fortran program

```
PROGRAM GOT_IT_WRONG
IMPLICIT NONE
INTEGER:: A, B, C = 3, D(1:5)
COMPLEX:: E = (/ 2, 4/)
B=C**3
A=SQRT(B*C)
DO COUNT=1,5
    D(COUNT-1)=B-COUNT*A
END DO
E=E*(0,1)
IF (D(4) < D(3)) THEN PRINT, E
END PROGRAM GOT_IT_WRONG
```

   (d) What will be printed when the following DO loops run?

(i)
```
REAL:: COUNT
LOGICAL:: TRUTH
DO COUNT=1,7,3
    TRUTH=((COUNT**2-8*COUNT+7) >= 0)
    PRINT *,TRUTH
END DO
```

(ii)
```
INTEGER:: OUTER, INNER, P
DO OUTER=2,5,3
    DO INNER=3,1,-1
      P=3*(OUTER/INNER)-(3*OUTER)/INNER
      PRINT *,P
    END DO
END DO
```

2. Write a FORTRAN program to print out to the screen a logical truth table for the following two logical statements

   STATEMENT1=A and (B or C)

   STATEMENT2=(A and B) or (A and C)

   (note that FORTRAN syntax differs somewhat from the above).

   Remember to declare all necessary logical variables. Using a DO loop, or otherwise, ensure that the program covers all possibilities for the logical inputs A, B, C. The program should test whether or not STATEMENT1 and STATEMENT2 are equivalent and print out the result.

   Write out the expected output from your program.

3. (a) The Euler method for solving the first order ordinary differential equation

$$\frac{dy}{dx} = F(x, y)$$

   is given by

$$x_{n+1} = x_n + h, \qquad y_{n+1} = y_n + hF(x_n, y_n).$$

   Briefly describe a more accurate method for solving this equation.

   (b) Describe how the Euler method can be used to solve the second order differential equation

$$\frac{d^2y}{dx^2} = G\left(x, y, \frac{dy}{dx}\right).$$

   (c) Write a FORTRAN program to find $y(1)$ using the Euler method if

$$\frac{d^2y}{dx^2} - x^2\frac{dy}{dx} + 2y = 0, \qquad y(0) = 0, \quad \frac{dy}{dx}(0) = 1.$$

   Let the step size $h = 0.01$, and write out the output y_1, y_2,...., y_{100} to a file 'SecondorderODE.dat'.

4. (a) Find a polynomial that interpolates the set of points $\{(-1,5),(1,-3),(4,10)\}$.

(b) Write down an iterative scheme based on the secant method that may be used to find a root of the equation

$$\tan x = e^{-x}.$$

(c) Describe the drawbacks associated with using the secant method to find a specific root of the above equation. Describe another algorithm that could be used with confidence to find the specific root lying between 2π and 3π. In general, does the secant method have an advantage over the method you have just described?

(d) The random variable X is uniformly distributed on [0,1]. Find a function $Y(X)$ to generate a random variable Y with density function

$$P(Y)\,dY = 2Ye^{-Y^2}\,dY, \qquad 0 < Y < \infty.$$

5. (a) Explain briefly the concept of a recursive subroutine or function in FORTRAN.

(b) In the standard (three post) version of the game 'Tower of Hanoi' the object is to move a set of different sized disks from the first post to the third. The disks are initially stacked in order of decreasing size on the first post, and must be moved one at a time without ever placing a large disk on a smaller one. The second post can be used to store disks whilst they are being moved.

(i) Describe how the problem of moving N disks from the first to third posts may be expressed in terms of operations involving moving $N-1$ or fewer disks.

(ii) Explain in words or FORTRAN statements how a recursive subroutine may be used to solve the 'Tower of Hanoi' using the algorithm described in (i). (You may assume the existence of a FORTRAN subroutine that may be used to move a single disks from one peg to another.)

(c) Write a recursive function in FORTRAN to evaluate the function $H(n)$, which is conjectured (but not proved!) to be defined for all positive integers n, and is given by

$$H(n) = \begin{cases} 0 & \text{for } n = 1 \\ H\left(\frac{n}{2}\right) + 1 & \text{for } n \text{ even} \\ H(3n+1) & \text{for } n \text{ odd.} \end{cases}$$

Evaluate $H(2)$ and $H(3)$.

6. A UCL department uses the following data type to store information about UK towns and cities

```
TYPE TOWN
  CHARACTER(LEN=20)::NAME
  REAL::COORDS(2)
  REAL::POPULATION
END TYPE TOWN
```

In each record the character string NAME holds the name of the town, the real array COORDS contains the eastward and northward distances in kilometres of the town from UCL, and POPULATION holds the town's population.

(a) Write a FORTRAN program which prompts the user for the name of a town and then searches an array UKTOWNS(10000), of type TOWN, for a town with the same name.

(b) The department uses the following formula to model (crudely) the number of people N travelling between two towns on a given day

$$N = \frac{\sqrt{P_1 P_2}}{(R+10)^2},$$

where P_1 and P_2 are the populations of the two towns, and R is the distance between them in kilometres. Write down the necessary FORTRAN commands, including the required function, to overload the operator $*$ so that the FORTRAN command

```
N=UKTOWNS(I)*UKTOWNS(J)
```

returns the number of people travelling between those towns with record indices I and J.