

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 : **11-MAY-04**

TIME : **10.00**

TIME ALLOWED : **2 Hours**

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 10.875?
(ii) What is the base 2 representation for 1/15?

- (b) (i) If N is the number of bits in a gigabyte, what is $\log_2(N)$?
(ii) Estimate very approximately how much memory is required to store a 1000×1000 pixel color photograph. Show details of your working.
(iii) How many bytes are needed to store the array
REAL :: STOREROOM(-2:2,0:1) if REAL numbers are stored in 32-bit floating point form.

- (c) Find FIVE errors in the following code:

```
PROGRAM ERRORPRONE
IMPLICIT NONE
INTEGER :: X,Y,Z
REAL :: P(X)
DO COUNT=1,4
  X=COUNT+3
  Y*Z=X+1
  P(X)=Y*Z
  IF (X==7) THEN P(X)=COUNT**2
  END IF
END DO
WRITE(*,6) X,Y
END PROGRAM ERRORPRONE
```

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

- (i) REAL:: A, INDEX
DO INDEX=5,-5,-3
 A=2**INDEX
 PRINT *,A
END DO
- (ii) INTEGER :: INNER, OUTER
DO OUTER =1,4
 DO INNER =1,3,2
 PRINT *,OUTER/INNER
 END DO
END DO

2. (a) Write a function SERIESEXP that calculates an approximate value for e^x from the first N terms its Taylor series expansion. The function should be written so that it accepts a complex argument and returns a complex value. The first few lines might be

```
FUNCTION SERIESEXP(XVALUE,N)
  COMPLEX, INTENT(IN):: XVALUE
  INTEGER, INTENT(IN):: N
  COMPLEX:: SERIESEXP
```

Care should be taken to minimise truncation errors when summing the series (you may assume that N will be chosen to be sufficiently large that the final terms in the series will be small).

- (b) Using SERIESEXP or otherwise, write functions SERIESSIN and SERIESCOS to approximate $\sin y$, $\cos y$ for real y , using terms in the series expansion up to y^n .
- (c) Write a main program using SERIESSIN and SERIESCOS to numerically verify De Moivre's theorem

$$(\cos y + i \sin y)^m = \cos my + i \sin my$$

for $m = 2, \dots, 10$. The program should prompt the user for y and N , and then write out to the screen a table comparing the calculated values of the left hand side and right hand side for each value of m .

3. (a) A function $f(x)$ is stored at a set of evenly spaced points $x_i = x_0 + ih$, for $i = 0, 1, \dots, N$. Write down an approximation for df/dx at x_i in terms of $f(x_{i+1})$ and $f(x_i)$. For the function $f(x) = e^x$, estimate the error to leading order in h .
- (b) Using the above approximation for first order derivatives, derive an algorithm to solve numerically the initial value problem involving the second order differential equation

$$\frac{d^2 f}{dx^2} + \left(\frac{df}{dx}\right)^2 + f = 0, \quad f(0) = 1, \quad \frac{df}{dx}(0) = 0.$$

(c) Using Lagrange polynomials, or otherwise, find a polynomial that interpolates the set of points $\{(1, 3), (2, 2), (4, 6)\}$.

(d) The random variable X is uniformly distributed on $[0,1]$.

(i) Describe (using FORTRAN commands if desired) how to use X to generate a random variable Z that has the same probability distribution as the total obtained when two uniform dice are rolled.

(ii) Use X to generate a random variable $Y(X)$ that has density function

$$P(Y) dY = \frac{2}{\pi(Y^2 + 1)} dY, \quad 0 < Y < \infty.$$

4. (a) By considering the tangent to the curve at x_n , or otherwise, show that the Newton-Raphson iteration for finding a root of the continuously differentiable function $f(x)$ is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

- (b) Consider the cubic polynomial

$$f(x) = x^3 + ax^2 + bx + c.$$

Write a FORTRAN program to read in real constants a , b and c and output to the screen

- (i) The turning points of $f(x)$ (i.e. points where $f'(x) = 0$, $f''(x) \neq 0$)
 (ii) The number of roots of $f(x)$.

(You may neglect cases where turning points and roots coincide).

- (c) In the case where $f(x)$ has one root and no turning points, write a subroutine to make N iterations towards finding the root from a starting point $x_0 = 0$, and returning the answer in RESULT. The subroutine might begin

```
SUBROUTINE SINGLEROOT(A,B,C,N,RESULT)
  REAL, INTENT(IN) :: A,B,C
  INTEGER, INTENT(IN) :: N
  REAL, INTENT(OUT) :: RESULT
```

- (d) Describe briefly how the subroutine SINGLEROOT could be adapted for the case where there is a single root but two distinct turning points.
 (Hint: consider making restrictions on the starting position x_0 for the iteration.)

5. Write a FORTRAN program to verify the epsilon-delta relation

$$\epsilon_{ijk}\epsilon_{klm} = \delta_{il}\delta_{jm} - \delta_{im}\delta_{jl}$$

for all possible inputs $i, j, l, m \in \{1, 2, 3\}$. (Note that there is an implied summation over all possible values of k on the left-hand side.) Here δ is the Kronecker delta, and ϵ is the Levi-Civita function, defined respectively by

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad \epsilon_{ijk} = \begin{cases} 1 & \{i,j,k\} \text{ a positive permutation of } \{1,2,3\}, \\ -1 & \{i,j,k\} \text{ a negative permutation of } \{1,2,3\}, \\ 0 & \text{otherwise.} \end{cases}$$

Good programming would involve defining functions to evaluate δ and ϵ , and using appropriate loops over all possible values of the indices. The program should write out to the screen the index values and the corresponding outputs for the left and right sides of the equation. By keeping track of whether the left and right hand sides are equal, the program should declare at the end whether the relation is correct.

6. Consider the data type

```
TYPE CUSTOMER
  CHARACTER(LEN=20):: FIRSTNAME
  CHARACTER(LEN=20):: LASTNAME
  REAL:: EXPENDITURE
END TYPE CUSTOMER
```

A retail outlet uses this data type to keep track of its customer records (first and last names and total expenditure), storing them in an array of size N (N is the number of customers).

(a) Write a subroutine `FINDCUSTOMER` to search the records for a customer's last name given in the input `MATCHNAME`. When the subroutine finds a matching record, it should write to the screen the full name of the customer and the amount of their expenditure. If no matching records are found, the subroutine should return 'No records found.' The subroutine might begin

```
SUBROUTINE FINDCUSTOMER(RECORDS, MATCHNAME)
  TYPE(CUSTOMER), INTENT(IN):: RECORDS(:)
  CHARACTER(LEN=20), INTENT(IN):: MATCHNAME
```

(b) Write a subroutine `SORTSPENDERS` to arrange the records in order corresponding to the amount spent by each customer. This subroutine might begin

```
SUBROUTINE SORTSPENDERS(UNSORTED, SORTED)
  TYPE(CUSTOMER), INTENT(IN):: UNSORTED(:)
  TYPE(CUSTOMER), INTENT(INOUT):: SORTED(:)
```

(c) Describe, in words or FORTRAN statements, a **fast** algorithm for searching the ordered array for a customer who has spent exactly 62 pounds (you may assume there is only one such customer).