COMPUTER SCIENCE TRIPOS Part II (General)
DIPLOMA IN COMPUTER SCIENCE

Thursday 6 June 2002 1.30 to 4.30

Paper 13 (Paper 4 of Diploma in Computer Science)

Answer **five** *questions.*

Submit the answers in five **separate** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* **all** *attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

1 Data Structures and Algorithms

Arithmetic encoding compactly represents a string of characters by an enormously precise number in the range [0,1) represented in binary by a finite sequence of digits following the decimal point. What is remarkable is that this number can be processed efficiently using only fixed point arithmetic on reasonably small integers. As a demonstration, if the original text contained only the characters `A`, `B`, `C` and the end-of-file mark `w`, such text can be arithmetically encoded using only 3-bit arithmetic. Illustrate how it can be done by decoding the string `101101000010` on the assumption that the character frequencies are such that the decoding tables of size 8 and 6 are, respectively, `wAABBCCC` and `wABBCC`. The first few lines of your working could be as follows:

```
                        0   0   0   0   1   1   1   1
                        0   0   1   1   0   0   1   1
                        0   1   0   1   0   1   0   1
101 101000010         |-w---A---A---B---B-+(C)++C+++C+|       => C
```

Your answer should include a brief description of how the decoding algorithm works. [20 marks]

2 Computer Design

(*a*) What is *microcode* and how does it differ from assembler? [6 marks]

(*b*) In assembler, branch instructions are used to change the flow of control. How can flow control be determined in a microcode environment? [4 marks]

(*c*) With the aid of a diagram, explain what a *feedback path* (sometimes called a *bypass*) is and how it is used to improve the throughput of a pipeline. [6 marks]

(*d*) What is a *branch delay slot*? [4 marks]

3 Digital Communication I

Define a *resource* in a digital communication system as anything whose use by one instance of communication prevents simultaneous use by another. Channel capacity is one example.

(*a*) Give *two* more examples of resource in digital communication systems. [4 marks]

(*b*) For the three resources, indicate how the amount of total resource can be increased. [6 marks]

(*c*) How are allocations of each of these resources to instances of communication performed? [10 marks]

4 Business Studies

(*a*) Explain the differences between

(*i*) credit and debit;

(*ii*) cash-flow and profit & loss statements;

(*iii*) equity and debt finance;

(*iv*) NPV and IRR;

(*v*) asset and DCF based valuation.

[2 marks each]

(*b*) A certain small software company has assets of about £100K (not including development work-in-progress), and an average cash-flow of about £15,000 per month, with a net profit of around £2000 per month. They are developing, but have not yet completed, a new graphical search engine into which they have invested about £100K of design and programmer time. The founders have invested about £150K, mostly in equity, and there is a long term debenture of £100K.

Provide a range of valuations for the company. Include notes explaining your assumptions and the basis for each valuation. [10 marks]

5 Comparative Programming Languages

(*a*) Briefly discuss the compromises that must be made when standardising a programming language. [8 marks]

(*b*) Discuss the relative merits to (1) the application programmers and (2) compiler writer of the following ways of specifying a programming language.

(*i*) A concise readable user manual for the language in English containing many useful programming examples.

(*ii*) A very long and highly detailed description, in English, of every feature of the language. This manual contains no programming examples.

(*iii*) A concise but rigorous description using a formal grammar to describe the language syntax and making extensive use of mathematical notations taken from set theory, λ-calculus, predicate calculus and logic to describe the semantics of the language.

(*iv*) The source code for a clean and elegant machine-independent interpretive implementation of the language.

[12 marks]

6 Compiler Construction

Explain a possible implementation technology for Java classes and objects. Your answer should focus on storage layout for objects and on how class variables and methods are accessed—it is not necessary to explain access qualifiers such as `public` and `private`. Illustrate your answer with the following program; in particular indicate its eventual output.

```
class test {
public int n;
public static int s = 100;
public void f(int x) { System.out.println("f1 " + (x+n)); }
public static void main(String args[]) {
        test p = new test();
        test2 q = new test2();
        test r = q;
        p.n = 4;
        q.n = 5;
        q.m = 6;
        r.n = 7;
        p.f(p.s);
        q.f(p.s);
        r.f(q.s);
     }
}

class test2 extends test {
public int n, m;
public static int s = 200;
public void f(int x) { System.out.println("f2 " + (x+n+m)); }
}
```

[20 marks]

7 Prolog for Artificial Intelligence

A simple D-type flip-flop is represented by the Prolog predicate `dff` whose definition is as follows:

```
dff(D, 0, Q, Q).
dff(D, 1, Q, D).
```

The first argument is the input to the flip-flop, the second is the clock with `0` representing a falling edge and `1` representing a rising edge. The third and fourth arguments are the previous and next states of the flip-flop. As can be seen the state of the flip-flop changes on a rising edge of the clock.

A clocked circuit consists of three d-type flip-flops with inputs and states (D_1, Q_1), (D_2, Q_2) and (D_3, Q_3). They are wired in such a way that

$$\begin{aligned} D_1 &= (Q_1 \wedge Q_2) \vee (\overline{Q_1} \wedge \overline{Q_2}) \\ D_2 &= (\overline{Q_1} \wedge Q_3) \vee (Q_2 \wedge \overline{Q_3}) \\ D_3 &= (Q_1 \wedge Q_3) \vee (\overline{Q_2} \wedge \overline{Q_3}) \end{aligned}$$

(*a*) Using `s(Q1, Q2, Q3)` to represent the state of the circuit, define a predicate that will compute the state after the next rising edge of the clock. You may find it helpful to define predicates to represent *and*, *or* and *not* gates. [14 marks]

(*b*) Define a predicate `testcc(N, s(Q1,Q2,Q3), List)` that will compute the list of states (`List`) through which the circuit passes from the given initial state `s(Q1,Q2,Q3)` as a result of a sequence of `N` rising edges of the clock. [6 marks]

8 Databases

(*a*) Define the operators in the (core) relational algebra. [6 marks]

(*b*) The core relational algebra is often extended with other operators. For the following operators give a definition and an example of their behaviour:

(*i*) the full outer join operator; [3 marks]

(*ii*) the aggregate and grouping operator. [5 marks]

(*c*) X, Y and Z are all relations with a single attribute A. A naïve user wishes to compute the set-theoretic expression $X \cap (Y \cup Z)$ and writes the following SQL query.

```
SELECT X.A
FROM   X,Y,Z
WHERE  X.A=Y.A OR X.A=Z.A
```

(*i*) Give the relational algebra term that this query would be compiled to. [2 marks]

(*ii*) Does the SQL query satisfy the user's expectation? Justify your answer. [4 marks]

9 Numerical Analysis II

Consider the alternative formulae

$$y_{n+1} = y_n + hf(x_n, y_n) + O(h^2) \tag{1}$$

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) + O(h^3) \tag{2}$$

applied to the ODE

$$y' = -5y, \quad y(0) = 1$$

using $h = 0.1$ in each case.

(*a*) Define the terms *local error* and *order* for an ODE formula. What is the *order* of each of the methods (1) and (2)? [2 marks]

(*b*) Giving answers to 2 significant decimal digits of accuracy, compute the solution of the ODE for $x_n = 0, 0.1, 0.2, \ldots$ 1.0 for each method. Tabulate your answers. The exact solutions to 2 significant digits are:

1.0, 0.61, 0.37, 0.22, 0.14, 0.082, 0.050, 0.030, 0.018, 0.011, 0.0067

Assume the exact value of $y(0.1)$ for method (2). [7 marks]

(*c*) Which method is more accurate initially and why? Explain the behaviour of each method as x increases. [3 marks]

(*d*) Solve the ODE. Find a general term for y_n in method (1) and show that the absolute error in (1) will be small when n is large. Without performing any further calculations, how do you expect the absolute error in method (2) to behave when n is large? [5 marks]

(*e*) Discuss briefly the suitability of formulae (1) and (2) as predictors for predictor–corrector methods in respect of *order* and *stability.* [3 marks]

10 Introduction to Functional Programming

(*a*) Write an ML function `merge` of type

```
('a * 'a -> bool) -> ('a list * 'a list) -> 'a list
```

which takes a comparison function of type `'a * 'a -> bool` and gives a function for merging two lists of type `'a` according to this function. [6 marks]

(*b*) Use your function `merge` to write a curried polymorphic function `mergesort` which takes a comparison function f and yields a sorting function of the appropriate type. [6 marks]

(*c*) Write a function `sumcomp` which takes two integer lists and returns the boolean value `true` if the sum of integers in the first list is no greater than the sum of integers in the second list and `false` otherwise. [6 marks]

(*d*) What is the type of the expression `mergesort sumcomp`? [2 marks]

11 Computer Vision

(*a*) Consider the “eigenfaces” approach to face recognition in computer vision.

(*i*) What is the rôle of the database population of example faces upon which this algorithm depends? [4 marks]

(*ii*) What are the features that the algorithm extracts, and how does it compute them? How is any given face represented in terms of the existing population of faces? [4 marks]

(*iii*) What are the strengths and the weaknesses of this type of representation for human faces? What invariances, if any, does this algorithm capture over the factors of perspective angle (or pose), illumination geometry, and facial expression? [4 marks]

(*iv*) Describe the relative computational complexity of this algorithm, its ability to learn over time, and its typical performance in face recognition trials. [4 marks]

(*b*) What is the following block of code doing over the image array `image[i][j]` as it computes the resulting new image array `result[i][j]` ? Give the appropriate mathematical name for this operation, and describe what it accomplishes. What are some computer vision tasks that might use this block of four nested `for loops`?

```
for (i = 0; i < iend; i++) {
    for (j = 0; j < jend; j++) {
        sum = 0;
        for (m = 0; m < mend; m++) {
            for (n = 0; n < nend; n++ ) {
                sum += kernel[m][n] * image[i-m][j-n];
            }
        }
        result[i][j] = sum/(mend*nend);
    }
}
```

[4 marks]

12 Complexity Theory

(*a*) State and prove the time hierarchy theorem. [10 marks]

(*b*) For each of the following statements, state whether or not it can be derived as a consequence of the time hierarchy theorem. Give justification for your answer.

(*i*) There is a language in $\textsc{Time}(n^2)$ that is not in $\textsc{Time}(n \log n)$. [3 marks]

(*ii*) There is a language in $\textsc{Time}(2^n)$ that is not decidable in polynomial time. [4 marks]

(*iii*) There is a language in $\textsc{Time}(2^n)$ that is not in **NP**. [3 marks]

END OF PAPER