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

Tuesday 3 June 1997 1.30 to 4.30

Paper 11 (Paper 2 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.*
Write on **one** *side of the paper only.*

1 Software Engineering I

Describe the *waterfall model* of software development and discuss its strengths and weaknesses. [12 marks]

For which of the following projects would it be suitable, and why?

(a) an incremental compiler for Java

(b) a clinical-record-keeping system for dentists

(c) a word-processing package

(d) a guidance system for an interplanetary probe

[8 marks]

2 Modula-3

The secretary of a health club stores the names, ages and heights of the club's clients in a database and requires a Modula-3 program to sort the clients three ways: into alphabetical order of name, into ascending order of age and into ascending order of height.

An early version of the program contains the following declaration of a variable `p`:

```
VAR
  p := ClientArray {
         NEW (RefClient, name := "JACK", age := 34, height := 1.70),
         NEW (RefClient, name := "JILL", age := 22, height := 1.76),
         .
         .
         .
                  };
```

Provide suitable `TYPE` declarations for the identifiers `RefClient` and `ClientArray`. [4 marks]

The heading of the required sort procedure is:

```
PROCEDURE Sort (VAR p : ARRAY OF RefClient; fges : CompType) =
```

where the type of the formal parameter `fges` is declared as:

```
  CompType = PROCEDURE (VAR p : ARRAY OF RefClient;
                        i : CARDINAL) : BOOLEAN;
```

The procedure `Sort` may be called in three ways:

```
  Sort (p, NameComp);
  Sort (p, AgeComp);
  Sort (p, HeightComp);
```

which sort the clients by name, age and height respectively.

Write a suitable procedure `AgeComp` which compares the ages of two adjacent clients. [4 marks]

Write a suitable body for the procedure `Sort` which uses a simple sort-by-exchange algorithm that uses the procedure argument `fges`. [12 marks]

3 Further Modula-3

Describe the use of generic interfaces and implementations in Modula-3. [8 marks]

Why do generic modules have interfaces as parameters rather than types? [2 marks]

A Modula-3 library is required to model a set of values of some (generic) type. For illustration, a set could be modelled as an opaque object with methods to initialise it as an empty set, to insert a value of the element type, to test a value for membership in the set and to form a new set as the union with another set.

Write a generic interface for this set type. [5 marks]

Sketch an implementation including a concrete revelation of the type and signatures of the default methods but omitting their actual code. Indicate the kinds of constraints that might be imposed when instantiating the module. [5 marks]

4 Distributed Systems

For a distributed filing system

(*a*) discuss the placement within an overall architecture of

(*i*) pathname resolution [3 marks]

(*ii*) existence control; [3 marks]

(*b*) discuss the need for an authentication infrastructure for such a system and outline how this might be provided; [8 marks]

(*c*) discuss how access control might be built above an authentication infrastructure. [6 marks]

5 Data Structures and Algorithms

At frequent intervals 100,000 numbers arrive in a message. It is required to select and record their median and throw the rest away.

Describe and justify the algorithm you would use to do this as quickly as possible, and explain, as if to a manager, any risks you are taking. [6 marks]

If the manager insisted that the risk be either (a) very greatly reduced or even (b) eliminated, outline what you would do. [6 marks]

In case (b) explain to the manager why he is almost certainly misguided and then describe in detail the technique you would adopt. [8 marks]

6 Operating System Foundations

Outline the components of interrupt handling and, where appropriate, discuss the tradeoffs of implementation by hardware or by software. [7 marks]

Give *two* examples of machine instructions on which mutual exclusion can be built. [2 marks]

Explain why temporarily suppressing interrupts can be used as a basis on which to build mutual exclusion for a uniprocessor but not for a multiprocessor. [2 marks]

For a uniprocessor describe how interrupt control, together with scheduling policy, can be used to protect writeable data shared by processes and hardware-driven routines. [3 marks]

Outline how user-level (top–down) requests for I/O can be synchronised with hardware-driven I/O software. [6 marks]

7 Operating System Functions

In the management of virtual memory, what is *thrashing*, and how does it occur? [5 marks]

What is the *working set* of a process, and how can it be computed? [5 marks]

List *five* techniques that can be used in an operating system to improve the performance of demand paged virtual memory. [5 marks]

What is a *capability*, and how can it be used for access control in a computer system? [5 marks]

8 Mathematics for Computation Theory

The events E, E' over finite alphabets S, S' are recognised by deterministic finite automata M, M' respectively. Show from first principles that there is a deterministic finite automaton $M_\cap$ that recognises the intersection $(E \cap E')$ of the two events. [10 marks]

Let S be the finite alphabet $\{a, b, c\}$.

(a) Let E be the event over S consisting of those words which contain an odd number of occurrences of a. Construct a deterministic finite automaton to recognise precisely the event E. [4 marks]

Write down a regular expression that denotes E. [2 marks]

(b) Let F be the event over S consisting of those words which contain an even number of occurrences of each of the symbols a, b, c. Show that F is regular. [4 marks]

9 Computation Theory

Define what is meant by saying that a set of *partial recursive* (μR) functions is *recursively enumerable*. Explain briefly how the universal register machine might be used to define a universal μR function $\mu(e, x)$ that enumerates the set of *all* partial recursive functions of a single variable x. [6 marks]

(a) Prove that the set of all total recursive functions of a single variable is not recursively enumerable. [4 marks]

(b) Show that there are recursively enumerable sets that are not recursive. [6 marks]

(c) Show that there is a partial recursive function that cannot be extended to any total recursive function. [4 marks]

[Any properties of recursively enumerable sets that you assume should be clearly stated.]

10 Numerical Analysis I

Explain the terms *unit round off* and *machine epsilon* (*macheps*). Why is *machine epsilon* used in preference to *unit round off* for practical purposes? [4 marks]

In the IEEE binary floating-point Standard (*IEEE* 754), what *exponent* and *significand* are used in representing each of the numbers 0, 1 and 2 in single precision? How are the exponent and significand stored in each case? [6 marks]

Show the 32 bits that represent (1+*macheps*). What is the *exact* value of *macheps* in this case? [4 marks]

What are the two sources of error in the formula

$$f'(x) \simeq \frac{f(x+h) - f(x)}{h}$$

and how does each type of error behave as h increases? [4 marks]

Suggest a suitable value of h if using this formula with IEEE single precision when $f(x) = O(1)$. [2 marks]

11 Computer Graphics and Image Processing

Describe an algorithm to draw a straight line using only integer arithmetic. You may assume that the line is in the first octant, that the line starts and ends at integer coordinates, and that the function *setpixel*(x, y) turns on the pixel at location (x, y). [13 marks]

Explain how straight lines can be used to draw Bezier cubic curves. [7 marks]

12 Digital Electronics

A lock has five buttons and a knob used to open the lock when the five buttons have been pressed in the correct sequence. Whenever the knob is turned, successfully or not, the state of the lock is reset. The correct sequence is hardwired into the control logic of the lock.

(*a*) Describe the inputs and outputs for a logic circuit which can be used to control the lock. [5 marks]

(*b*) Draw a state diagram for the logic circuit. [10 marks]

(*c*) Derive equations for the logic system using J-K flip flops to hold state variables. *Do not minimise the equations.* [5 marks]

13 Natural Language Processing

The following dialogue between a hypothetical User, U, and a database access system, S, illustrates some of the problems inherent in natural language processing by machine.

U: Is every type of shock absorber stocked in one warehouse in any city?
S: Yes, in the South Coventry warehouse.
U: Which types are held in Birmingham?
S: sa55 and sa65 shock absorbers for Metros and Montegos.

(*a*) Describe the problems illustrated by this extract. [6 marks]

(*b*) Give a sketch of an architecture of a natural language interface to a database which could support such exchanges. [6 marks]

(*c*) Say how, and to what extent, the problems described could be solved within such an architecture. [8 marks]