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

Tuesday 6 June 1995 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.*
Write on **one** *side of the paper only.*

1 Modula-3

Define the Modula-3 terms *positional parameter* and *named parameter* and write a program fragment which illustrates the difference. [5 marks]

Write a procedure `FmInD` which is designed to format an `INTEGER` to a specified (`CARDINAL`) field width, returning the result as `TEXT`. To take a simple case, `FmInD(-246,7)` would return `-246` as `TEXT` but padded with three leading spaces to make a field width of 7 characters.

If the specified field width is too small, as in `FmInD(-246,3)`, then the minimum sensible width is used (`4` in this case). If no field width is specified, a default value of `8` is assumed.

The only library procedure which you may use is `Fmt.Char` which converts type `Char` to type `TEXT`. The procedure should be carefully annotated to explain how it works.

[15 marks]

2 Further Modula-3

Describe the facilities provided in Modula-3 for abstraction through object inheritance and partial revelations of opaque types. [8 marks]

Illustrate your answer by reference to the Trestle toolkit, mentioning the relationship between `VBT.T`, `VBT.Leaf`, `VBT.Split`, `Filter.T` and their sub-types. [4 marks]

Explain how overriding methods such as mouse, position and keyboard in a sub-type of `VBT.T` can be used to define a new class of window. [4 marks]

How is initialisation of super-types handled? [4 marks]

3 Software Engineering

Describe the particular difficulties involved in constructing very large software systems, as observed by Curtis *et al.* List some of the root causes of these difficulties. [7 marks]

Write an essay on the rôle of testing in a software project. The essay should describe testing's purpose and limitations, and touch upon black-box, white-box and acceptance testing, ultra-high reliability and beta-testing. (A mere list of definitions will receive little credit.) [13 marks]

4 Compiler Construction

Outline how minimum cost code can be compiled from a parse tree using rules that consist of tree template replacements, costs and corresponding code. Illustrate your answer by considering the translation of the expression

```
Add(Add(K1, K2), Add(K3, K4))
```

using the following rules:

```
#1  Ri  <-  Add(Ri, Rj)            cost: 2   code: ADDR  Ri,Rj

#2  Ri  <-  Add(Ri, Kc)            cost: 3   code: ADDI  Ri,c

#3  Ri  <-  Kc                     cost: 2   code: LOADI Ri,c

#4  Ri  <-  Add(Ri, Add(Rj, Kc))   cost: 4   code: ADDRI Ri,Rj,c
```

In your answer you should derive the finite state machine needed for the efficient matching of these four rules, and you should also give the cost and resulting translation of the given example expression. [15 marks]

Briefly discuss in what ways this algorithm may fail to generate optimum code when used in a compiler. [5 marks]

5 Data Structures and Algorithms

What is a *priority queue*? Explain the data structure known as a *heap* and document how a heap is stored in a simple linear block of memory. [4 marks]

If a heap stores N items, describe how it can be viewed as an almost-balanced binary tree. What difference can there be between the greatest and least lengths of paths from the root of the tree to a leaf? What operations must be performed to move from one node in the tree to (a) its parent and (b) its offspring? [5 marks]

Describe, and estimate the costs of, procedures to

(a) insert a new item into an existing heap;

(b) delete the topmost item from a non-empty heap;

(c) starting from an array holding N items in arbitrary order, rearrange those items so that they form a heap, taking time less than that which would be needed if the items were just inserted into the heap one after the other. [6 marks]

A *stable* sorting method is one where items whose keys compare as equal will appear in the output in the same order that they appeared in the input list. Would a heap sort based on the algorithms you have documented be stable? Justify your answer. [5 marks]

6 Topics in Artificial Intelligence

Give some reasons why some tasks are suited to being solved by using *constraints*. Illustrate your answer with examples of such tasks and the methods by which constraints are used to solve them. [20 marks]

7 Operating System Foundations

Consider the operation of a scheduler in a system where there are system level and user level processes. User processes may be IO bound or CPU bound and may have user controlled (negative) priority.

Describe the data structures that the scheduler might use, including parts of process descriptors that the scheduler would operate on. [10 marks]

Describe in detail the circumstances under which the scheduler would be entered and for each different circumstance outline a scheduling algorithm that might be used. [10 marks]

8 Operating System Functions

An operating system for a *smart card* is required — that is a common run-time system for the smart card, which hosts some set of applications loaded into the card at the time the card is issued.

What operations should the memory management component of the operating system support in such a system? [12 marks]

The hardware engineers are concerned about complex memory management hardware consuming too much power, but the customer demands memory protection between the applications. Describe some solutions that could meet these demands. [8 marks]

9 Graphics

Explain the purpose and operation of the A-buffer in rendering a sequence of images into a framestore. [12 marks]

Exhibit an example that shows an advantage over the use of a Z-buffer. [8 marks]

10 Computation Theory

Explain what is meant by a *primitive recursive* function and by a *partial recursive* function. [6 marks]

Show that the function giving the next state of a register machine in terms of the current state is primitive recursive. (You may assume the existence of primitive recursive functions for coding any n-element list of numbers $(x_1, \ldots, x_n)$ as a number $[x_1, \ldots, x_n]$ (for each n), and for extracting the head x_1 and the (coded) tail $[x_2, \ldots, x_n]$ from such a coded list.) [8 marks]

Deduce that every register machine computable partial function is partial recursive. [5 marks]

Is the converse true? [1 mark]

11 Numerical Analysis I

The Newton–Raphson formula for solution of $f(x) = 0$ is

$$\hat{x} = x - \frac{f(x)}{f'(x)}.$$

By means of a sketch graph, describe how the method works in a simple case. [4 marks]

When the method converges, what *rate of convergence* is expected? Describe one circumstance in which the method may fail to converge. [4 marks]

Consider the simultaneous equations

$$\left.\begin{aligned} f_1(x_1, x_2) &= x_2 - {x_1}^2 - 2 = 0 \\ f_2(x_1, x_2) &= x_1(x_2 - 3x_1) = 0 \end{aligned}\right\} \tag{1}$$

Suppose the iterative scheme

$$\begin{pmatrix} -2x_1 & 1 \\ x_2 - 6x_1 & x_1 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} -f_1(x_1, x_2) \\ -f_2(x_1, x_2) \end{pmatrix} \tag{2}$$

is applied to the equations (1). If $\{x_1, x_2\}$ is the starting approximation, the improved approximation is given by

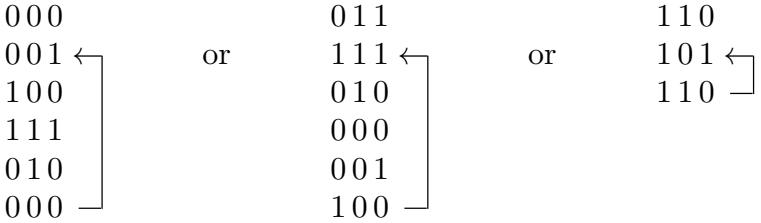
$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}.$$

Suppose $x_1 = 0$. Show, by solving the equations (2) that the first iteration always produces the same improved approximation for any non-zero x_2. [10 marks]

Verify that the method converges if x_1 is set to 0, and $x_2 \neq 0$. [2 marks]

12 Digital Electronics

A sequential circuit has been designed which behaves slightly erratically. When switched on it produces on its three output wires one of the following patterns:



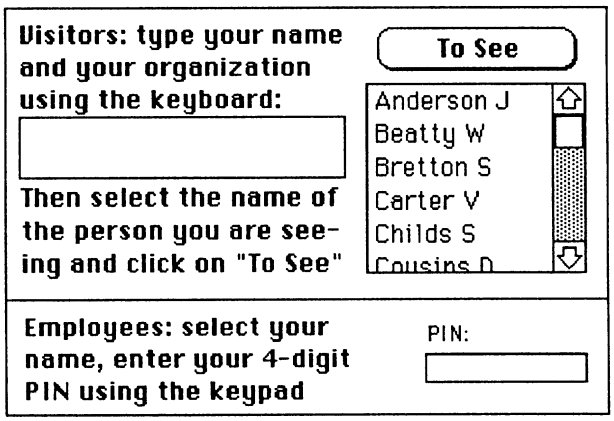
The last pattern was not intended to arise, although the other two are acceptable. Deduce a possible circuit which exhibits this behaviour. [12 marks]

Propose a modification to your circuit which eliminates the third pattern. [8 marks]

13 Designing Interactive Applications

At the reception desk of a highly secure organisation, all visitors and staff are required to register before entry to the building. The organisation's management is concerned to reduce registration delays without increasing levels of staffing at the desk.

(a) Write a one-sentence statement of a problem in interactive system design, aimed at addressing this situation. [3 marks]



(b) The figure above shows the screen layout of a possible system for visitor and staff registration, using multiple terminals, each equipped with display, keyboard, keypad and mouse. Name three methods you might use in assessing the usability of this design, and the reasons why each method might be useful. [5 marks]

(c) Describe, for the benefit of a new member of the design team, how to go about applying *one* of the three methods you have named in part (b). [8 marks]

(d) What interaction style, other than the direct-manipulation style illustrated in the figure, would be appropriate for this application,