COMPUTER SCIENCE TRIPOS Part IA

Monday 4 June 2001 1.30 to 4.30

Paper 1

Answer **two** *questions from Section A, and* **one** *question from* **each** *of Sections B, C, D and E.*
Submit the answers in six **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.*

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

SECTION A

1 Foundations of Computer Science

(a) An ML program makes the following declarations:

```
val x = ref 0;

fun f n = (x := !x + 1;  n + !x);

fun g n =
  let val x = ref 0
  in  x := !x + 1;  n + !x  end;
```

Consider evaluating each of the following expressions:

```
map f [1,2,3,4];
map g [1,2,3,4];
map ref [5,5,5];
```

What value is returned in each case and how are the references affected? [5 marks]

(b) Code the function `filter` such that `filter p xs` returns the list of those elements of the list `xs` satisfying the predicate `p`. [1 mark]

(c) Use `filter` to express Quicksort. [4 marks]

2 Discrete Mathematics

(a) Prove the fundamental theorem of arithmetic, that any natural number can be expressed as a product of powers of primes and that such an expression is unique up to the order of the primes. [4 marks]

(b) Given a natural number n, let $d(n)$ be the number of divisors of n (including 1 and n).

If $p_1, p_2, \ldots, p_k$ are distinct primes, prove that

$$d({p_1}^{\alpha_1} {p_2}^{\alpha_2} \ldots {p_k}^{\alpha_k}) = \prod_{i=1}^{k} (\alpha_i + 1).$$ [3 marks]

(c) What is the smallest number with 36 factors? [3 marks]

3 Java

Explain how the word "protected" is used in Java, commenting about when and why a programmer might use it rather than the alternative legal words that can appear in similar places. Why will you hardly ever use "protected" in small programs that you write and run for yourself? [10 marks]

4 Operating Systems

For *each* of the following, indicate whether the statement is true or false, and explain why this is the case (no marks will be awarded for an answer with no explanation).

(a) Round-robin scheduling can suffer from the so-called "convoy effect".

(b) System calls are an optional extra in modern operating systems like Windows 2000.

(c) A paged virtual memory is smaller than a segmented one.

(d) In UNIX, hard-links cannot span mount points.

(e) Direct memory access (DMA) makes devices go faster.

[2 marks each]

SECTION B

5 Foundations of Computer Science

(a) Write brief notes on the following:

(i) Re-coding a function to make it iterative. [4 marks]

(ii) The difference between depth-first and breadth-first search. [4 marks]

(iii) ML's exception-handling facilities. [4 marks]

(b) Code a function whose input is a list of integers and whose output is the list of all the integers that can be expressed as the sum of one or more of the supplied integers. For example, given [1,5,10] a correct output is [1,5,10,6,11,15,16]; the order of the elements in the output does not matter. [5 marks]

(c) Modify the function that you coded above so that the elements of its output appear in numerical order and without repetitions. [3 marks]

6 Foundations of Computer Science

To represent the power series $\sum_{i=0}^{\infty} a_i\, x^i$ in a computer amounts to representing the coefficients a_0, a_1, a_2, ... One possible representation is by a function of type `int->real` that returns the coefficient a_i given i as an argument. An alternative representation is the following datatype:

```
datatype power = Cons of real * (unit -> power);
```

(a) Demonstrate the two representations by using each of them to implement these two power series:

(i) The constant power series c, with $a_0 = c$ and $a_i = 0$ for $i > 0$. [3 marks]

(ii) The Taylor series $\sum_{i=0}^{\infty} x^i/i!$ for the exponential function. [4 marks]

(b) Also implement (using both representations) each of the following operations on power series:

(i) Product with a scalar, given by $c \cdot \left(\sum_{i=0}^{\infty} a_i\, x^i\right) = \sum_{i=0}^{\infty} (c a_i)\, x^i$. [3 marks]

(ii) Sum, given by $\left(\sum_{i=0}^{\infty} a_i\, x^i\right) + \left(\sum_{i=0}^{\infty} b_i\, x^i\right) = \sum_{i=0}^{\infty} (a_i + b_i)\, x^i$. [4 marks]

(iii) The product $\left(\sum_{i=0}^{\infty} a_i\, x^i\right) \times \left(\sum_{i=0}^{\infty} b_i\, x^i\right)$, where the ith coefficient of the result is $a_0\, b_i + a_1\, b_{i-1} + \cdots + a_i\, b_0$. [6 marks]

You may assume there is an ML function `real` of type `int->real` that maps an integer to the equivalent real number.

SECTION C

7 Discrete Mathematics

(*a*) State carefully the Fermat–Euler theorem, defining any terms that you use. [4 marks]

(*b*) Explain how calculating $a^{n-1} \pmod{n}$ for various values of a can be used to show that n is composite without actually finding its factors. By considering $561 = 3 \times 11 \times 17$ or otherwise, show that the test is not perfect and suggest an improvement to make it more selective. [6 marks]

(*c*) Derive the RSA system for public key cryptography and explain how this can be used both to send messages that are kept secret from an interceptor and to prove the identity of a sender. [6 marks]

(*d*) Show that knowledge of the secret key as well as the public key allows an interceptor to factor the modular base being used. [4 marks]

8 Discrete Mathematics

Let $(\mathrm{A}, \leqslant_{\mathrm{A}})$ and $(\mathrm{B}, \leqslant_{\mathrm{B}})$ be partially ordered sets.

(*a*) Define the *product order* on $\mathrm{A}\times\mathrm{B}$ and prove that it is a partial order. [4 marks]

The *upper bound* of a set $\mathrm{S} \subseteq \mathrm{A}$ is an element $u \in \mathrm{A}$ (but not necessarily in S) such that $\forall\, s \in \mathrm{S} \,.\, s \leqslant u$. The *least upper bound* of S is an upper bound of S that is less than every other upper bound of S. The *greatest lower bound* is defined similarly.

A *lattice* is a partially ordered set in which every pair of elements has both a least upper bound and a greatest lower bound.

(*b*) Prove that $(\mathbb{N}, |)$, the natural numbers under the divisibility order, form a lattice. [4 marks]

(*c*) Given a set X, prove that $(\mathcal{P}(\mathrm{X}), \subseteq)$, the power set of X under set inclusion, forms a lattice. [4 marks]

(*d*) Does every subset of $(\mathbb{N}, |)$ have a least upper bound and a greatest lower bound? Justify your answer. What about $(\mathbb{N}_0, |)$ and $(\mathcal{P}(\mathrm{X}), \subseteq)$? [4 marks]

(*e*) If $(\mathrm{A}, \leqslant_{\mathrm{A}})$ and $(\mathrm{B}, \leqslant_{\mathrm{B}})$ are lattices, show that $\mathrm{A}\times\mathrm{B}$ is a lattice under the product order. [4 marks]

SECTION D

9 Programming in Java

For each of the following areas, write brief comments on the way in which Java and its libraries have been designed to try to prevent programmers from making undetected errors and to ensure that code runs on all possible brands and models of computer, yielding the same results in each case.

(*a*) Programmers who get mildly confused about syntax or who make typing errors. [4 marks]

(*b*) Groups of programmers working on libraries that will form part of some large project. [4 marks]

(*c*) Numbers, characters, trig functions and so on. [4 marks]

(*d*) Opening files, reading or writing and then closing them. [4 marks]

(*e*) Other features of Java not falling centrally within any of the above categories. [4 marks]

10 Programming in Java

For each of the following ML definitions write, explain and comment on the closest reasonable Java equivalent code-fragment that you can construct.

(*a*)
```
fun fact n r = if n = 0 then r
                  else fact (n-1) (n*r);
```
[4 marks]

(*b*)
```
datatype 'a tree = leaf of 'a |
          node of 'a tree * 'a tree * 'a tree;
```
[4 marks]

(*c*)
```
val v = [1,2,3,7];
val v2 = map (fn n=>n*n:int) v;
```
[4 marks]

(*d*)
```
exception Disaster of int;
raise Disaster 99;
```
[4 marks]

(*e*)
```
fun recursiveLength nil = 0
  | recursiveLength (a :: b) = 1 + recursiveLength b;
```
[4 marks]

SECTION E

11 Operating Systems

(*a*) Describe how the CPU is allocated to processes if static priority scheduling is used. Be sure to consider the various possibilities available in the case of a tie. [4 marks]

(*b*) "All scheduling algorithms are essentially priority scheduling algorithms."

Discuss this statement with reference to the first-come first-served (FCFS), shortest job first (SJF), shortest remaining time first (SRTF) and round-robin (RR) scheduling algorithms. [4 marks]

(*c*) What is the major problem with static priority scheduling and how may it be addressed? [4 marks]

(*d*) Why do many CPU scheduling algorithms try to favour I/O intensive jobs? [2 marks]

(*e*) Describe how this is achieved in the (*i*) UNIX and (*ii*) Windows NT operating systems. [3 marks in each case]

12 Operating Systems

(*a*) From the point of view of the device driver, data may be read from an I/O device using *polling*, *interrupt-driven programmed I/O*, or *direct memory access* (DMA). Briefly explain each of these terms, and in each case outline using pseudo-code (or a flow chart) the flow of control in the device driver when reading data from the device. [14 marks]

(*b*) From the point of view of the application programmer, data may be read from a device in a *blocking*, *non-blocking* or *asynchronous* fashion. Using a keyboard as an example device, describe the expected behaviour in each case. [6 marks]

END OF PAPER