

# COMPUTER SCIENCE TRIPOS Part IA

---

Monday 2 June 2003 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.*

**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) Carefully explain the concept of a curried function, with examples. [4 marks]

(b) Explain the purpose and operation of the function `f`:

```
fun foldr f ([], e) = e
  | foldr f (x::xs, e) = f(x, foldr f (xs,e));
```

```
fun f xs =
  foldr (fn ((x,y),(xs,ys)) => (x::xs,y::ys)) (xs, ([], []));
```

[4 marks]

(c) Write a function that accepts a list and returns the list consisting of its first, second, fourth, fifth, seventh, eighth, tenth, etc., elements. In other words, your function should return all the elements except those whose position in the input list is a multiple of three. [2 marks]

### 2 Discrete Mathematics

(a) Show that the highest common factor of 798 and 567 is 21. [2 marks]

(b) Find all pairs of integers  $(x, y)$  with  $798x + 567y = 63$ . [4 marks]

(c) Find all integer solutions to  $567x = 42 \pmod{798}$ . [4 marks]

### 3 Programming in Java

Examine the following statements and indicate which are true and which are false. In most cases a simple one-word answer “true” or “false” will suffice, but, if you feel the need to make assumptions about circumstances or otherwise elaborate a little, please keep your notes as short as possible.

- (a) All methods in an abstract superclass must be declared abstract.
- (b) A class declared final cannot be subclassed. Every method of a final class is implicitly final.
- (c) A redefinition of a superclass method in a subclass need not have the same signature as the superclass method. Such a redefinition is not method overriding but is simply an example of method overloading.
- (d) A constructor is a special method with the same name as the class that is used to initialise the members of a class object. Constructors are called when objects of their classes are instantiated.
- (e) A method declared static cannot access non-static class members. A static method has no this reference because static class variables and static methods exist independent of any objects of a class.
- (f) An array subscript may be an integer or an integer expression. If a program uses an expression as a subscript, then the expression is evaluated to determine the particular element of the array.
- (g) To pass one row of a double-subscripted array to a method that receives a single-subscripted array, simply pass the name of the array followed by the row subscript.
- (h) Overloaded methods can have different return values, and must have different parameter lists. Two methods differing only by return type will result in a compilation error.
- (i) The applet’s paint method is called after the init method completes execution and the start method method has started executing to draw on the applet. It is also called automatically every time the applet needs to be repainted.
- (j) Any block of Java code may contain variable declarations.

[1 mark each]

## 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) Preemptive schedulers require hardware support.
- (b) The Unix shell supports redirection to the buffer cache.
- (c) A context switch can be implemented by a flip-flop stored in the translation lookaside buffer (TLB).
- (d) Non-blocking I/O is possible even when using a block device.
- (e) Shortest job first (SJF) is an optimal scheduling algorithm.

[2 marks each]

## SECTION B

### 5 Foundations of Computer Science

- (a) Describe how lazy lists can be implemented using ML. [2 marks]
- (b) Code a function to concatenate two lazy lists, by analogy to the ‘append’ function for ordinary ML lists. Describe what happens if your function is applied to a pair of infinite lists. [3 marks]
- (c) Code a function to combine two lazy lists, interleaving the elements of each. [3 marks]
- (d) Code the lazy list whose elements are all ordinary lists of zeroes and ones, namely  $[], [0], [1], [0, 0], [0, 1], [1, 0], [1, 1], [0, 0, 0], \dots$  [6 marks]
- (e) A *palindrome* is a list that equals its own reverse. Code the lazy list whose elements are all palindromes of 0s and 1s, namely  $[], [0], [1], [0, 0], [0, 0, 0], [0, 1, 0], [1, 1], [1, 0, 1], [1, 1, 1], [0, 0, 0, 0], \dots$ . You may take the reversal function `rev` as given. [6 marks]

## 6 Foundations of Computer Science

- (a) Explain in detail how a binary search tree represents a dictionary. If reference types are not used, what is the average-case cost of an update operation? [8 marks]
- (b) A *mutable* binary tree is either a leaf or a branch node containing a label and references to two other mutable binary trees. Present the ML datatype declaration for mutable binary trees. [2 marks]
- (c) Describe how to use this datatype to implement (mutable) binary search trees. Give ML code for the lookup and update operations. The update operation must never copy existing tree nodes. [10 marks]

## SECTION C

### 7 Discrete Mathematics

- (a) Prove that there are infinitely many prime numbers. [4 marks]
- (b) Let  $p_1, p_2, \dots, p_k$  be the first  $k$  primes. Show that the number of positive integers less than  $n$  and having no prime factors other than  $p_1, p_2, \dots, p_k$  is less than  $\sqrt{n}2^k$ . [8 marks]

[Hint: All such numbers are of the form  $m^2 p_1^{\varepsilon_1} p_2^{\varepsilon_2} \cdots p_k^{\varepsilon_k}$  where each  $\varepsilon_i$  is 0 or 1.]

Deduce that the  $k^{\text{th}}$  prime is less than  $4^k$ . [4 marks]

- (c) State the inclusion–exclusion principle and use it to determine the number of prime numbers less than 100. [4 marks]

## 8 Discrete Mathematics

- (a) Define the terms *injective*, *surjective* and *bijective*, and state the Schröder–Bernstein theorem concerning the existence of a bijection between two sets. [4 marks]
- (b) What is a countable set? [2 marks]
- (c) Prove the following assertions:
- (i) If  $C$  is a countable set and  $f : A \rightarrow C$  is an injection, then  $A$  is countable. [2 marks]
- (ii) If  $A$  and  $B$  are countable sets, then  $A \times B$  is countable. [2 marks]
- (iii)  $\mathbb{Z}$  and  $\mathbb{Q}$ , the sets of integer and rational numbers, are both countable. [4 marks]
- (iv)  $\mathcal{P}(\mathbb{N})$ , the set of all subsets of the natural numbers, is not countable. [2 marks]
- (v) If  $U$  is an uncountable set and  $f : U \rightarrow V$  is an injection, then  $V$  is not countable. [2 marks]
- (vi)  $\mathbb{R}$ , the set of real numbers, is not countable. [2 marks]

**SECTION D****9 Programming in Java**

Explain the following Java words. For each write a sentence giving an overview of what the word is about, then either write a second sentence that shows how it fits in with related concepts or write a small code-fragment to illustrate its purpose:

- (a) `abstract`;
- (b) `assert`;
- (c) `boolean`;
- (d) `break`;
- (e) `byte`;
- (f) `case`;
- (g) `catch`;
- (h) `extends`;
- (i) `new`;
- (j) `null`.

[2 marks each]

Please keep your explanations of each word brief so that your whole answer to this question remains a reasonable size: you are not expected to produce a detailed explanation or comprehensive illustration.

## 10 Programming in Java

You are to write a Java program. Its main class should be called `Replicate` and its behaviour is to be that the following two sequences of Unix commands must result in exactly the same text appearing on your terminal:

1. `cat Replicate.java`
2. `javac Replicate.java`  
`java Replicate`

in other words the output from the program should be identical to its own source. Your code is not permitted to inspect the file system, so items along the lines of

```
... new InputStream(new File("Replicate.java")) ...
```

are prohibited. Your code is expected to be reasonably neatly laid out and must contain at least one comment! Credit will be given for code brevity and for inclusion of a clear explanation of how your code works.

[20 marks]

[Hint: it is very probable that your code will consist of two rather similar sections. One will be data and the other executable code. They will be such that the code can print the data out in two different ways: once to re-create itself and a second time to re-create the code.]

## SECTION E

### 11 Operating Systems

- (a) Describe, with the aid of diagrams where appropriate, how Unix implements and manages:
- (i) a hierarchical name space for files; [2 marks]
  - (ii) allocation of storage on disk; [2 marks]
  - (iii) file-system and file meta-data; [8 marks]
  - (iv) pipes. [4 marks]
- (b) A system administrator decides to make a ‘versioned’ file-system in which there are a number of directories called `/root-dd-mm-yyyy`, each of which holds a copy of the file-system on day `dd`, month `mm` and year `yyyy`. The idea is that at any particular time only the most recent snapshot will be used as the ‘real’ filesystem root, but that all previous snapshots will be available by explicitly accessing the directory in question. In this way the system administrator hopes to allow resilience to mistaken edits or unintentional deletions by users, or to hardware problems such as a disk head crash.

To implement this, the system administrator arranges for a program to run every morning at 01:00 which recursively ‘copies’ the current snapshot to the new one. However to save disk space, hardlinks are used in place of actual copies. Once the ‘copy’ is complete, the new snapshot is used as the new root.

To what extent will this scheme provide the functionality the system administrator hopes for? What advantages and disadvantages does it have? [4 marks]

## 12 Operating Systems

- (a) Describe with the aid of a diagram how a simple computer executes a program in terms of the *fetch-execute cycle*, including the ways in which arithmetic instructions, memory accesses and control flow instructions are handled. [12 marks]
- (b) Explain how a program accesses I/O devices when:
- (i) it is running in supervisor-mode; [5 marks]
  - (ii) it is running in user-mode. [3 marks]

**END OF PAPER**