

QUEEN MARY, UNIVERSITY OF LONDON

DCS128 ALGORITHMS AND DATA STRUCTURES

Class Test Monday 4th April 2005 10.35-11.55

Please fill in your Examination Number here:

Student Number here:

All answers to this test should be written on the test sheet, but you may use spare paper for rough working. Answer as many of the questions as you can.

- 1) Explain in the space below what an **ordered binary tree** is, and explain informally how it is used to maintain a set of objects.

- 2) In this question you can assume the implementation in Java of an abstract data type called `List`, which stores a collection of integers and has the following operations given by non-static methods:
- `head` – returns the first item of the list
 - `tail` – returns a new list consisting of all the items from the list except the first item
 - `cons` – takes an integer argument and returns a new list whose `head` is that integer and whose `tail` is the list the `cons` operation was called on.
 - `IsEmpty` – returns `true` if the list it is called on is the empty list, `false` otherwise.
- There is also a static method called `empty` which returns a new `List` object representing the empty list.

The code you write should not assume anything about `List` objects except what is given above.

- a) Write a Java static method `replace` which takes as its argument a `List` of integers and two integers and returns a `List` of integers in which all occurrences of the first integer argument have been replaced by the second. For example, if the textual representation of the `List` is `[5, 7, 9, 3, 7, 4, 1, 2, 8, 7, 3]` and the integer arguments are 3 and 6, the `List` constructed has the textual representation `[5, 7, 9, 6, 7, 4, 1, 2, 8, 7, 6]`.

- b) Write in the space below a static method which takes two `List` objects as arguments and returns a `List` object which contains all the integers of the first `List` argument which also occur in the second `List` argument. The order of the integers in the `List` returned does not matter. You may define and use an auxiliary method if you need to.
- For example, if the two `List` arguments have textual representations `[3,10,4,7,8,12,6]` and `[5,4,12,9,8,13,3]` the `List` object returned will have the textual representation `[3,4,8,12]` or something with the same integers in a different order.

3) In the space below give a Java implementation of the abstract data type Queue , using a linked list representation with an extra pointer to the last cell.

- 4) a) Describe briefly the difference between **pre-order** and **post-order** tree traversal.
- b) Explain briefly the technique which is used to obtain **breadth-first** tree traversal.
- c) Explain, without using code, the technique used to **join** one linked list to the end of another **destructively**.
- d) Explain, without using code, the “**stack and reverse**” technique for copying a linked list.

- 5) Describe **three** different data structures that may be used to implement the abstract data type “sequence” (that is, a list with an insertion point). In each case explain briefly how the data structure represents the abstract data type.