QUEEN MARY, UNIVERSITY OF LONDON
DCS/128 ALGORITHMS AND DATA STRUCTURES

Class Test Monday 25th March 2002 15.05-15.55

All answers to this test should be written in the separate answer book provided.

Answer as many of the questions as you can.

1) Explain the algorithm for deleting an item from an ordered binary tree.

2) Write a Java class which implements the abstract data type Queue.

3) Given an implementation of the abstract data type List in a Java class `List.java`, with non-static methods `head`, `tail`, `cons`, `isempty`, implementing the equivalent operations, static method `empty` which produces a new empty list, and no public constructors, write static methods to implement the following operations. In both cases, assume your implementation implements a list of characters.

   a) `takeTill` takes a list and a character and returns that portion of the list which occurs before the first occurrence of the character. So if the list is [a,b,c,d,b,d,e,f] and the character is d, the method should return a representation of the list [a,b,c].

   b) `changeNth` takes a list, two characters, and an integer `n`, and returns the list resulting from changing the `n`th occurrence of the first character in the list to the second. So if the list is [a,b,c,d,b,d,e,f,d,g,d], the characters are d and x, and `n` is 3, the list returned is [a,b,c,d,b,d,e,f,x,g,d].

4) Assume we have a class `Table` which stores items of type `Thing` in a table. Amongst the methods available in class `Table` are those with signatures:

```
public Enumeration getEnumeration()
public void addThing(Thing t)
```

The first method returns an enumeration which can be used to go through all the items in the table. The second method adds a new item to the table. A zero-argument constructor for `Table` produces a new table with no entries.

Suppose in class `Tester` we have a method with signature:

```
public static boolean test(Thing t)
```

Your task is to write a static method to go in class `Tester` which takes a `Table` object as its argument and returns a new `Table` object which contains all those items from the original `Table` object where a call of `test` with them as the argument returns true.