SOLUTION NOTES

Foundations of Computer Science 2002 Paper 1 Question 5 (LCP)

This question assesses basic ML programming skills, and specifically lecture 7 (Datatypes and Trees) and Lecture 11 (List Functionals). The solutions given are particularly concise through the use of functionals such as `map`, but other working solutions are acceptable.

(a)−−(b) Here are `flip` and `paint`:

```
fun flip (Wave(x,fs)) = Wave(x, rev (map flip fs));

fun paint f (Wave(x,fs)) = Wave(f x, map (paint f) fs);
```

(c) Here is one approach to `same_shape`, reproducing functions that were presented in the course:

```
fun zip (x::xs,y::ys) = (x,y) :: zip(xs,ys)
  | zip _             = [];

fun forall pred []      = true
  | forall pred (x::xs) =
        (pred x)  andalso  forall pred xs;

fun same_shape (Wave(_,fs1), Wave(_,fs2)) =
    length(fs1) = length(fs2) andalso
    forall same_shape (zip(fs1,fs2));
```

(d) Candidates should be able to guess the types even if they can't write the code.

```
flip : 'a fan -> 'a fan
paint : ('a -> 'b) -> 'a fan -> 'b fan
same_shape : 'a fan * 'a fan -> bool
```

(e) This function counts the number of `Wave` nodes in a tree, or more precisely it adds that number to its argument `q`. Through `foldr`, the function is recursively applied to each subtree of a node from right to left. There is no need to form explicit lists of the counts arising from each subtree.