

# COMPUTER SCIENCE TRIPOS Part IA – 2007 – Paper 1

## 5 Foundations of Computer Science (LCP)

*This question has been translated from Standard ML to OCaml*

(a) Consider the following piece of OCaml code:

```
type 'a tree = Lf | Br of 'a * 'a tree * 'a tree
exception Blair

let rec tony p = function
  | Lf -> true
  | Br (x,t1,t2) -> if not (p x) then raise Blair
                    else try tony p t1 with Blair -> tony p t2

let gordon p t = try tony p t with Blair -> false
```

(i) Code a function that returns the same results as `gordon` but makes no use of exceptions. [4 marks]

(ii) What property of binary trees does `gordon` express? [3 marks]

(b) Write brief notes on the OCaml type `exn`. [3 marks]

(c) Consider the following piece of OCaml code:

```
type 'a result = Ian of 'a | Cherie of exn

let what f x = try Ian (f x) with e -> Cherie e
```

We ask OCaml to evaluate the expression

```
map (what (tony (fun x -> x <> 0))) [ta; tb]
```

and the response is as follows:

```
- : bool result list = [Ian true; Cherie Blair]
```

What is the type of `what (tony (fn x => x <> 0))`, and what can we infer about the binary trees `ta` and `tb`? Justify both answers carefully. [5+5 marks]