UNIVERSITY OF LONDON
(University College London)

B.Sc. DEGREE 1998

COMPUTER SCIENCE B11A: INTRODUCTORY PROGRAMMING I

Answer THREE Questions.

The Use of Electronic Calculators: is NOT Permitted.

Answer Question 1 and any TWO other questions.

You may find the following definitions useful throughout this paper:

```
id x = x

lay []       = ""
lay (a:x)    = a ++ "\n" ++ (lay x)

map f []     = []
map f (x:xs) = (f x) : (map f xs)
```

1. (a) Explain the terms “lazy evaluation” and “list comprehension”. [4]

   (b) What value does the following expression compute?

```
last [(x ^ 2) + 1 | x <- [1..]; (x ^ 2) < 5000000]
where
last []     = error "last of empty list"
last [x]    = x
last (x:xs) = last xs
```

   [6]

   (c) What is a type? [2]

   (d) Why will Miranda not allow the following function definition?

```
dumb x = (x x)
```

   [6]

   (e) Given the following function definitions, where `number` is the name of a type synonym:

```
one :: number
one f x = f x

two :: number
two f x = f (f x)

three :: number
three f x = f (f (f x))
```

   — Give the most general type synonym definition for `number`. [4]

   — What is the most general type of the following function?

```
operator a b = h
               where
               h c x = a c (b c x)
```

   [6]

   — Explain the operation of the function `operator` (in the context of the other definitions given above) by giving the evaluation steps of a simple application. Then explain in general terms what the function `operator` does (for example, could it be given a more descriptive name?). [6]

[Total 34]

[TURN OVER]

2. (a) What are algebraic data types? Give examples of the different kinds of algebraic type and how they might be used. [5]

(b) Define a type structure to represent binary trees in which the nodes of the tree hold number values and the leaves also hold number values. [4]

(c) Define a function to determine the height of a tree represented using your type, where the height of a tree is the number of nodes along the longest branch from the root to a leaf. [9]

(d) Consider the following function defined for lists:

```
> map_on_tails f [] = []
> map_on_tails f xs = (f xs) : (map_on_tails f (tl xs)
```

Define an analogous function on the trees represented by your type, where a function is applied to every sub-tree within a tree. [11]

(e) Define a function which will take a tree and return a tree containing at each node the height of the corresponding sub-tree in the input tree. [4]

[Total 33]

3. (a) Briefly explain, with examples, what is meant by the following terms:

partial application
case analysis
structural induction

[10]

(b) Briefly explain the advantages of using Higher Order Functions. Illustrate your answer by giving the code (and type) for a higher-order function (call it `gsort`) which will sort a list of any type of object in any user-provided ordering. [10]

(c) Discuss, with examples, the role played by *recursion* in functional programming. Your answer should address (amongst other things) the following points:

— Recursive function definitions.

— Recursive types (both built-in and user-defined).

— Stack, accumulative and mutual recursion. [13]

[Total 33]

[CONTINUED]

4. (a) Provide definitions, including types, for the two functions (from the Miranda Standard Environment) called `foldr` and `foldl`. [12]

(b) What values do the following five expressions compute?

```
foldr (:) [] [1,2,3]

hd (foldr (:) [] [1..])

foldl (:) [] [1,2,3]

foldl (swap (:)) [] [1,2,3]
where
swap f x y = f y x

foldl (swap (:)) [] [1..]
where
swap f x y = f y x
```

[5]

(c) Under what circumstances are the functions `foldr` and `foldl` interchangeable? [8]

(d) What does the following function do and what is its type?

```
f x = foldr rcons id x []
      where
      rcons a f b = f (a:b)
```

Demonstrate how f works by giving the intermediate evaluation steps of f applied to an argument. [8]

[Total 33]

[TURN OVER]

5. The game called “Minefield” presents the user with a 10x10 grid of cells that are initially blank. Five (5) of these cells contain hidden mines. The user is invited to enter (x,y) coordinates (such that both x and y are between 1 and 10 inclusive) for cells that she wishes to visit. The user is given one point for every cell that is visited that does not contain a mine. As soon as the user visits a cell that contains a mine, or visits a cell that has already been visited, the game is over and the program prints the user’s score to the screen.

   (a) You are given the following type synonym definition for the game board:
   ```
   >board == [[cell]]
   ```
   Give an algebraic type definition for the type `cell`. [2]

   (b) Give the definition (including its type) of the function `init_board` which takes as its argument a (possibly infinite) list of (x,y) coordinates representing the positions of the mines. Your function should generate a value of type `board` containing 95 empty cells and 5 cells containing mines in the appropriate positions. [19]

   (c) Give the definitions (including types) of the following two functions:

   — `usermove`. This function takes two arguments: the board and a single (x,y) coordinate. It returns a two-tuple containing (i) a boolean according to whether the user has hit a mine or a previously visited cell, and (ii) a new board, suitably updated to indicate which cells have been visited.

   — `showboard`. this function takes a single argument: the board. It returns a list of characters with newlines embedded appropriately in order to display the board on the screen. Previously-visited cells should be shown as the character ’X’ and unvisited cells should be shown as the space character. Mines should be shown as spaces (i.e. hidden from the user). [6]

   (d) Assuming the existence of a function `random` which produces a potentially-infinite list of random (x,y) coordinates, give the definition for the main function for the above game. This function should provide a loop so that before each move the screen is cleared and the current state of the board is displayed on the screen. The user is then invited to enter a new (x,y) coordinate pair, and this is checked with the function `usermove`. Loop again if no mine (or previously visited cell) is hit: otherwise, terminate and print out the user’s score. [6]

[Total 33]

[END OF PAPER]