Answer any THREE questions.

1. (a) Give the syntax for the untyped lambda calculus with operators and constants.

[8]

(b) Explain in detail how a lambda-calculus program is evaluated, including an explanation of the term “reducible expression”.

[11]

(c) Give a detailed explanation of the operation of the four reduction rules that can be used to evaluate an expression in the untyped lambda calculus with operators and constants.

[8]

(d) What is a combinator? Give lambda-calculus expressions for the three combinators known as **S**, **K**, and **I** (otherwise known as distribute, cancel and identity).

[6]

[Total 33]

[TURN OVER]

2. (a) Give a Miranda algebraic type definition for a binary tree called **numchartree** that contains one data value (only) in each node, and where that data value can be either a number or a character (but no other type). [4]

(b) Give the Miranda function definition, including its type, for a function called **insertnum** which inserts a number into a **numchartree** such that the tree contains only numbers and such that the numbers in the tree are *sorted*; i.e. for each node containing value **v**, all the numbers held in its right subtree are greater than or equal to **v**, and all the numbers held in its left subtree are less than **v**. [8]

(c) Modify your definition for **insertnum** so that when a new number is added to a **numchartree** the tree remains *balanced*. A balanced tree is one where, for each node in the tree, the numbers of nodes in its two subtrees differ by no more than one - i.e. if the number of nodes in the left subtree is x, and the number of nodes in the right subtree is y, then

```
abs (x-y) <= 1
```

[21]

[Total 33]

[CONTINUED]

3. You are given the following function definitions:

```
cancel x y = x

swap f x y = f y x

nil y = error "cannot take head or tail of empty list"

cons a b f = f a b

head x = x cancel

tail x = x (swap cancel)
```

(a) Use hand evaluation to demonstrate the following two equalities:

```
head (cons a b) = a

tail (cons a b) = b
```

[4]

(b) Use hand evaluation to demonstrate the following equality:

```
head (tail (cons a (cons b c))) = b
```

[8]

(c) Explain why the following function definition is wrong:

```
newmap f        nil = nil
newmap f (cons a b) = cons (f a) (newmap f b)
```

[6]

(d) Suggest a correct definition for the function **newmap** that maps a function over the elements of the “cons” as defined in this question (above). Do **not** atempt to give the type for **newmap**.

[15]

[Total 33]

[TURN OVER]

4. You are given the following definitions:

```
truth == (* -> * -> *)

true :: truth
true  = cancel

false :: truth
false = (swap cancel)
```

(a) Consider the functions **not**, **and**, and **or**. These are intended to mimic the standard Boolean operators of the same name, but they should work on values of type **truth** instead of values of type bool.

Give the Miranda type signatures for the functions **not**, **and**, and **or**. [6]

(b) Provide Miranda definitions for the functions **not**, **and**, and **or**, and demonstrate their correct working with example hand evaluations. [18]

(c) The Boolean function **xor** (exclusive or) returns true if either of its two arguments are true, but it returns false if either both arguments are false or both arguments are true.

Provide a Miranda definition (including the type) for the function **xor**, and demonstrate its correct working with an example hand evaluation. [9]

[Total 33]

[CONTINUED]

5. (a) State briefly what garbage collection is and why it is necessary for both Miranda and Java. Give a pictorial example of the creation of garbage in a graph reduction system.

[8]

(b) Describe briefly the operation of three different garbage-collection techniques and compare their advantages and disadvantages.

[15]

(c) What is fragmentation and how can it be cured? Your explanation should make reference to the three garbage collectors described in your answer to part (b) of this question.

[10]

[Total 33]

[END OF PAPER]