Answer Question 1 and any TWO other questions.

1. (a) What is a higher-order function? Give example function definitions in Miranda to illustrate two different kinds of higher-order function. For each function, provide an example application and give a step-by-step derivation to show how Miranda would evaluate that application.

[7]

(b) What is a recursive type? Give an example of a built-in recursive type in Miranda. How can you define your own recursive type? Give an example in Miranda of a user-defined type which is recursive.

[5]

(c) Give the syntax for the untyped lambda calculus without constants and give the definitions for the reduction rules that are used to evaluate expressions in that calculus..

[10]

[Question 1 cont. over page]
[TURN OVER]

[Question 1 cont.]

(d) Evaluate each of the following Miranda expressions. If you think that an expression gives an error, say why (if there is more than one error in an expression, explain all of them):

```
[ ] : [ ] : [ ]

([[ ]] : [[ ]]) : [[ ]]

[[ ]] : ([[ ]] : [[ ]])

((3-3)=0) & ((23 / 0) = 0)

((3-3)=0) \/  ((23 / 0) = 0)

((3-3)=0) \/  ((True / 0) = 0)

exp1
where
exp1 = 25, if (3 < 5 < 27)
     = False, otherwise

exp3 5
where
exp3 = id, if False
     = const 3, otherwise
```

[11]
[Total 33]

[CONTINUED]

2. “Functional Languages are ideally suited to the programming of parallel applications running on multiprocessor computers”. Discuss this statement with reference to the underlying evaluation mechanisms and properties of functional languages and the manner in which they may be implemented.

[Total 33]

3. (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 find 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. Define an analogous function on the trees represented by your type, where a function is applied to every sub-tree within a tree.

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

[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]

[TURN OVER]

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

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

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

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

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

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

hd (foldl (swap (:)) [] [1..])
```

[8]

(c) How do the functions `foldr` and `foldl` differ in their ability to process infinite lists?

[5]

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

```
mystery op = g
             where
             g r = (r:). rest
                   where
                   rest []    = []
                   rest (a:x) = g (op r a) x
```

[8]

[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]