UNIVERSITY COLLEGE LONDON

University of London

EXAMINATION FOR INTERNAL STUDENTS

For the following qualifications :-

B.Eng. *M.Eng.*

Comp Sci B330: Functional Programming

COURSE CODE	:	**COMPB330**
UNIT VALUE	:	**0.50**
DATE	:	**14-MAY-02**
TIME	:	**10.00**
TIME ALLOWED	:	**2 hours 30 minutes**

Answer any THREE questions.

1. (a) What is a recursive function? Give examples of function definitions in Miranda for both a stack recursive function and an accumulative recursive function. For each function, provide a sample function application and give the reduction steps (using any reduction order, to any normal form) for that application.

[7 marks]

(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 that is recursive.

[5 marks]

(c) Explain, with examples, what is meant by the following terms:

higher-order function

Currying

partial application

Explain how the above terms are related.

[10 marks]

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

[Question 1 cont.]

(d) Give the value computed by 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):

(i)
```
[ [ ] ]   :   [ [ [ ] ] ]
```

(ii)
```
[ [ [ ] ] ]   :   [ [ ] ]
```

(iii)
```
[ ] : [ ] : [ ]
```

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

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

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

(vii)
```
exp2 5
where
exp2 = 3, if True
     = 5, otherwise
```

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

[11 marks]
[Total 33 marks]

[CONTINUED]

2. (a) Provide an algebraic data type definition to represent the four suits in a deck of cards. Make use of this algebraic type in a type synonym to represent a particular card, for example "the ten of hearts".

[6 marks]

(b) Write a function that will take as arguments (i) a number and (ii) a list of elements. Each element in the list represents a card (using the types defined in (a)). The function should produce as its output a "shuffled" version of the input list. The function should shuffle the list of cards as many times as is indicated by the first argument.

The action of shuffling should cut the deck in half and then interleave the cards, with the previous top card now being the second card in the pack. For example, if a list of four items A, B, C and D is shuffled once the result should be C, A, D, B. If that result is shuffled again the result should be D, C, B, A. As a simplification, you may assume that where there are an odd number of items in the list, the last item should be discarded.

Your function should detect the case where there are more than 52 elements in the input list, which it should treat as an error.

[18 marks]

(c) Write a function which takes two integer numbers and generates a result using the following rules:

(i) Multiply the two numbers together

(ii) Then add all the digits of the result

(iii) If the sum of the digits has itself only one digit then return it as the result of the function, otherwise repeat from (ii)

For example, if your function is called "f", a sample interaction with the Miranda system would be:

```
Miranda f 3 4
3
Miranda f 7 7
4
Miranda f 30 19
3
```

The last of the above examples is calculated as follows:

*30 * 19 is 570;*
5 + 7 + 0 is 12 (which has 2 digits);
1 + 2 is 3 (which has 1 digit);
the result is 3.

[9 marks]
[Total 33 marks]

[TURN OVER]

3. (a) Given the definitions:

```
s f g x = f x (g x)

b f g x = f (g x)

w h x = h x x

e = s (b w b) (b w b)
```

and given that the Y fixpoint combinator is defined axiomatically as

```
Y f = f (Y f)
```

show that e is equivalent to Y by illustrating that partial reduction of the expression `(e f)` produces an expression that is equivalent to `(f (e f))`.

[12 marks]

(b) In fact the definition for e given above is not a valid Miranda definition. Explain why the definition of e gives rise to an error in Miranda.

[9 marks]

(c) What is the most general type of each of the following functions:

```
fun1 (a:b) f x = a ((x f b) . f)

fun2 []       j        acc = acc
fun2 f        []       acc = acc
fun2 (x : xs) (y : ys) acc = y   (fun2 xs ys (x y acc))


mysterious [] y z = z
mysterious (front : rest) y z
= (front z) & mysterious rest y ((y . front . y) z)
```

[12 marks]
[Total 33 marks]

[CONTINUED]

4. Given the following extract from a Miranda program::

```
|| Insertion sort program.
|| Elements of an unsorted list are inserted,
|| one at a time, into a sorted list.

isort :: [*] -> [*]
isort     [] = []
isort (x:xs) = insert x (isort xs)
               where
               insert x []     = [x]
               insert x (y:ys) = (x : (y:ys)), if (x < y)
                               = y : (insert x ys), otherwise
```

(a) Give Miranda's response for each of the following expressions assuming it is typed at the Miranda prompt:

```
isort ""

insert 2 []

(hd . isort) ["deck", "of", "cards"]

isort [isort,isort]
```

[6 marks]

(b) `isort` has type `[*] -> [*]`, yet the code for `isort` contains a call to `insert`, which contains the comparison `(x < y)`. Give examples of types other than num to which the relational operator < can be applied.

[4 marks]

(c) Briefly explain the advantages of using *Higher Order* functions. Illustrate your answer by generalizing `isort` to create a new *Curried* function `gsort`, which takes a first argument that is a function to be used in place of <, and a second argument which is the list to be sorted. This should permit isort to sort a list in any user-provided ordering. Give both the type declaration and function definition of `gsort`.

[8 marks]

(d) With reference to `isort` and `gsort` discuss the advantages of *polymorphic typing* and *Currying*.

[8 marks]

(e) Provide specialized versions of `gsort` as follows:

(i) To sort a list of strings in descending order.

(ii) To sort a list of 2-tuples of numbers in ascending order. For this function, (2,3) is considered less than (1,7), because (2+3) is less than (1+7).

[7 marks]
[Total 33 marks]

[TURN OVER]

5. (a) Provide definitions, including types, for the two functions (from the Miranda Standard Environment) called `foldr` and `foldl`. [10 marks]

(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 marks]

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

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

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

[10 marks]

[Total 33 marks]

[END OF PAPER]