

This test consists of four exercises. *Explain your answers.* The available points for each item are written in the margin.

[6]

1. Given the functions

```
let rec foldl f b xs = match xs with
  | []    -> b
  | x::xs -> foldl f (f b x) xs
```

```
let rec range m n = if m > n then [] else m :: range (m+1) n
```

evaluate the function call `foldl (fun ys x -> x :: ys) [] (range 1 2)` and give at least 6 intermediate steps.

[4]

2. (a) Implement a function `remdups : 'a list -> 'a list` that removes duplicate elements from a list. E.g.,

```
remdups [1;2;1;3] = [2;1;3]
```

Hint: The function `List.mem : 'a -> 'a list -> bool` may be useful.

[4]

(b) Implement a function `pair : 'a list -> ('a * 'a)list` with the following behavior:

```
pair [x1;x2;x3;x4;...;xn] = [(x1,x2);(x3,x4);...;(xn-1,xn)]
pair [x1;x2;x3] = [(x1,x2)]
```

[5]

3. Give the sets \mathcal{BVar} , \mathcal{FVar} , \mathcal{Var} , and \mathcal{Sub} for the λ -term $t = (\lambda bz.x a (y z)) (x y)$.

[6]

4. Rewrite the following λ -term to NF, giving all intermediate β -steps.

$$(\lambda mnfx.m f (n f x)) (\lambda fx.f x) (\lambda fx.x)$$