

SOLUTION NOTES

As submitted to the Examiners in January 2001.

Foundations of Functional Programming 2001 Paper 5 Question 10 (ACN)

The first 2 parts are essentially reproducing examples from the lectures. I will give them here in ML like syntax but the question REQUIRES that they be re-cast in lambda expression form.

```
fun Y f =  
  let g h = f (h h)  
  in g g
```

I can turn this into a lambda expresson in my head!

```
fun P x y = lambda f => f x y  
fun A p    = p (lambda x y => x)  
fun D p    = p (lambda x y => y)
```

The final part then says

```
let FG (f , g) = ( lambda x => (g (f (g x))),  
                  lambda x => (g (f x)) )  
then Y FG gives the pair (f, g)
```

However writing it PROPERLY in lambda form is more messy since I have to use A and D not pattern matching...

```
FG = lambda fg => P  
    (lambda x => ((D fg) ((A fg) ((D fg) x))) ... etc
```

But apart from being messy it is not really terribly hard or lengthy!
The last bit is not explicitly worked through in lectures.