1. (Functional Programming, 6pts)

(a) (2 pts) What is a closure?

A closure is a mapping from a function to a collection of variable bindings (i.e., an environment).

Consider the following OCaml function:

```ocaml
mult x y = x * y
```

(b) (4 pts) Show the evaluation of `mult 2 3`, including any closures.

```
mult 2 3 → <cl> 3 → 2 * 3 → 6
```

2. (OCaml Types and Type Inference, 6pts)

(a) (4 pts) Give the types of the following OCaml expressions

(i) `let f x = if x 3.14 else 9.81`  
   `TYPE = bool -> float`

(ii) `[[1;2;3];[4;5;6]]`  
    `TYPE = int list list`

(b) (2 pts) Write an OCaml expression with the type `int*int -> int`.

Some possible solutions:
```
fun (x,y) -> x+y
let f (x,y) = x*y
...  
```
3. (OCaml Programming, 8pts)
Using fold (given below) and an anonymous function, write a function `getMax` that, when applied to a list of ints `lst` will return the element with the maximum value.

Example: `getMax [2;5;39;12;46;22] = 46`

```ocaml
let rec fold f a l = match l with
  | [] -> a |
  | (h::t) -> fold f (f a h) t

let getMax = fold (fun a x -> if x > a then x else a) 0 OR
let getMax = fold (fun a -> fun x -> if x > a then x else a) 0
```