(a)
```
let curry g =
let f = (fun x -> (fun y -> g (x, y)))
f;;
val curry : g:('a * 'b -> 'c) -> ('a -> 'b -> 'c)

let uncurry f =
let g = (fun (x, y) -> f x y)
g;;
val uncurry : f:('a -> 'b -> 'c) -> ('a * 'b -> 'c)
```

(b) If the above functions are correct, then the following equalities should hold because the functions curry and uncurry are inverses. Let f be an uncurried function of type ('a * 'b − > 'c). To show that the functions are equivalent, apply the same (s, t) to both sides. Let s: 'a and t: 'b where f(s, t) = z : 'c.

```
uncurry(curry(f))(s, t) = f(s, t)
uncurry(f': ('a -> 'b -> 'c))(s, t) = f(s, t)
f'': ('a * 'b -> 'c)(s, t) = f: ('a * 'b -> 'c)(s, t)
z'' : 'c = z : 'c
```

This shows that the result of currying and uncurrying will be the same type as the original function, and thus the output of the same input will be of the same type. Because the curry and uncurry functions only change the way in which the arguments are applied (tuple instead of curried) but do not change the way the arguments are evaluated, then the output must be the same.

Because the functions are inverses, function composition in the reverse order should also evaluate to be equal to the original function. Let g be a curried function of type ('a − > 'b − > 'c). To show that the functions are equivalent, apply the same (s, t) to both sides. Let s: 'a and t: 'b where f s t = z : 'c.

To further show that uncurry and curry are inverse functions, let's look at two functions f and g that perform the same operation on two inputs: compare a string and an int and return a boolean that represents if the string length equals the integer.

```
curry(uncurry(g)) s t = g s t
curry(g': ('a * 'b -> 'c)) s t = g s t
(g'' : ('a -> 'b -> 'c)) s t = (g: ('a -> 'b -> 'c) s t
z'' : 'c = z : 'c
```

As shown above, g and g" are functions of the same type, and because the curry and uncurry functions only change the way in which and the order arguments are applied, it will not change the outcome of the function if given the same input. So the above equality is true.