Quiz #1

Swap 3 5
Swap 3 0 5
(5, 3)
(5, 3, 0)

Let swap int (x: int) (y: int) = (y, x)
Let swap real (x: float) (y: float) = (y, x)
Let swap int Real (x: Float) (y: int) = (y, x)
(5, Int) (y: Float)

Type polymorphism

Let swap (h(x: 'a) (y: 'b)) : b * x = (y, x)

Val swap : 'a * 'b -> 'b * 'a

Let append to string ((x: 'a), (s: string),
(convert: 'c -> string)) : String=
(convert x) ^ " " ^ s

'a * String & (a -> String) -> String
append to string (3110, "class", string of 110)
"3110 class"

Variant types

type answer = Yes | No | Maybe
CONSTRUCTORS

| type Either Point = TwoD of float * float      |
| | ThreeD of float * float * float              |

TwoD (3.1, 4.6)

let lastTwo (p : Either Point) : float * float =
match p with
| TwoD (x, y) -> (x, y)
| ThreeD (x, y, z) -> (y, z)

INT LIST

KONS

type intList = Nil | Cons of (int * intList)

Nil
Cons (1, Nil)
Cons (2, Cons (1, Nil))

let alist = Cons (1, Nil)
let blist = Cons (3, alist)

let rec length (clist : intList) : int =
match clist with
| Nil -> 0
| Cons (h, t) -> length (t) + 1

let rec is_empty (clist : intList) : bool =
match clist with
| Nil -> true
| Cons (_, _) -> false
let rec addone_all (l: int list): int list =  
  match l with  
    | Nil -> Nil  
    | Cons (h, t) -> Cons (inc h), addone_all t)  

let rec square_all (l: int list): int list =  
  match l with  
    | Nil -> Nil  
    | Cons (h, t) -> Cons (square h), square_all t)  

let rec do_f_to_all (f: int -> int, l: int list): int list =  
  match l with  
    | Nil -> Nil  
    | Cons (h, t) -> Cons (f h), do_f_to_all t)  

let square_all (l: int list): int list =  
  do_f_to_all (square, l)  

let rec sum (l: int list): int =  
  match l with  
    | Nil -> 0  
    | Cons (h, t) -> h + sum t