CIS 120 Quiz 2

January 28–29, 2015

Name: _____________________________________________________________

PennKey (e.g. mkizner): __________________________ Section: ____________

Indicate the section you’re registered for, even if you’re attending a different section.

```ml
type tree =
| Empty
| Node of tree * int * tree

let rec insert (v: int) (t: tree) =
begin match t with
| Empty -> Node (Empty, v, Empty)
| Node (lt, x, rt) ->
    if v < x then Node (insert v lt, x, rt)
    else if v > x then Node (lt, x, insert v rt)
    else t
end
```
1. Write the type of each of the following OCaml expressions in the blank provided, or ill-typed if the expression does not type check.

(a) let a : ____________________________ = [3 :: []]
(b) let b : ____________________________ = [(5, "hello"); (1, "")]  
(c) let c : ____________________________ = (1 :: 2) :: []
(d) let d : ____________________________ = (1 :: [2]) :: [3]
(e) let e : ____________________________ = [(5, 12, 7); (2, 5)]
(f) let f : ____________________________ = let (_, y) = (2, 58) in y

2. Write the value computed for \( r \) in the following code segment.

```ocaml
let rec f (l: int list) : int * int =  
begin match l with  
| [] -> (0, 0)  
| [x] -> (x, x)  
| x :: y :: tl ->  
  let (w, z) = f tl in  
  (x + w, y + z)  
end

let r = f [2; 3; 4; 5; 6]
```

*Answer:* \( r = \) ____________________________

3. Recall the definition of binary trees and of the `insert` function, reproduced on the front side of this page, and consider the following code segment.

```ocaml
let rec f (t1: tree) (t2: tree) : tree =  
begin match t1, t2 with  
| _, Empty -> t1  
| Empty, _ -> t2  
| _, Node (lt, x, rt) -> f (f (insert x t1) lt) rt  
end

let r1 = Node (Node (Empty, 4, Node (Empty, 7, Empty)),  
8, Node (Empty, 12, Empty))  
let r2 = Node (Node (Empty, 5, Empty), 7, Node (Empty, 11, Empty))  
let r = f r1 r2
```

*Answer:* r = ____________________________

Draw the tree \( r \) below.