CMSC330 Summer 2017 Quiz 2

Name (as it appears on Gradescope): ____________________________________________

Instructions:
- Do not begin this quiz until you are instructed to do so.
- You have 15 minutes to complete this quiz.
- This quiz is closed-book. No notes or other aids are allowed.
- Show all your work and clearly indicate your answers for partial credit.

1. [6 points] Give the types of the following OCaml expressions.
   a. [2 points] (1.0, [1 < 0])

   b. [2 points] fun x y -> x y

   c. [2 points] fun f (x, y) -> f x y

2. [4 points] Give OCaml expressions with the following types.
   a. [2 points] (string * int) list

   b. [2 points] ‘a list -> string * ‘a
let rec map f = function
  | []     -> []
  | x :: xs -> f x :: map f xs

let rec fold f v = function
  | []     -> v
  | x :: xs -> fold f (f v x) xs

type 'a option =
  | None
  | Some of 'a

3. [4 points] Evaluate the following OCaml expression. There are no syntax or type errors.

let rec foo xs ys =
  match (xs, ys) with
  | (x :: xs, y :: ys) -> (x, y) :: foo xs ys
  | _                -> [] in
fold (fun v (x, y) -> v +. x *. y) 0.0
  (foo [1.0; 0.5; 1.5] [1.0; 4.0; 2.0])

4. [6 points] Write a function map_options: ('a -> 'b) -> 'a option list -> 'b list that accepts as arguments a function f : 'a -> 'b and list of optional values xs : 'a option list and returns the result of applying f to each element of xs ignoring None values. You may use Pervasives.(@) or List.append. You must use fold or map.

map_options (fun x -> x +1) [None; None] = []
map_options (fun x -> x +1) [Some 1; None; Some 2] = [2; 3]