

# CMSC330 Fall 26 Quiz 2 - SOLUTIONS

Proctoring TA: \_\_\_\_\_ Name: \_\_\_\_\_

Section Number: \_\_\_\_\_ UID: \_\_\_\_\_

## Problem 1: OCaml Coding

[Total 20 pts]

You can use the following `fold_left`, `fold_right` and `map` functions:

```
let rec fold_left f a lst =
  match lst with
  [] -> a
  |x::xs -> fold_left f (f a x) xs

let rec fold_right f lst a =
  match lst with
  [] -> a
  |x::xs -> f x (fold_right f xs a)

let rec map f lst = match lst with
  [] -> []
  |x::xs -> (f x)::(map f xs)
```

(a) Length

[6 pts]

In lecture, we wrote a list length function with `fold_left`. Write that here:

(\* for reference \*)

let length lst =

```
let rec length lst = match lst with
  [] -> 0
  |x::xs -> 1 + (length xs)
```

```
fold_left (fun a _ -> a + 1) 0 lst
```

(b) Filtering

[14 pts]

Write a function that takes in an int list, and returns the value and its index if the value is odd. The order of the resulting list does not matter. You can use `fold_left`, `fold_right` and `map`, but otherwise, you **may not** use the `rec` keyword. You can assume that all ints will be non-negative.

(\* examples \*)

```
odd_and_indexes [10;20;37;40] = [(37,2)]
```

```
odd_and_indexes [19;12;13] = [(19,0);(13,2)]
```

```
odd_and_indexes [2;4;6] = []
```

Note: Tuple should be odd value then index

```
let odds_and_indexes lst =
```

(\* for reference \*)

```
let odds_and_indexes lst =
```

```
let rec helper i lst = match lst with
  [] -> []
```

```
|x::xs -> if x mod 2 = 0 then
```

```
  helper (i+1) xs
```

```
else
```

```
  (x,i)::(helper (i+1) xs)
```

```
in helper 0 lst
```

```
let ret,_ = fold_left
  (fun (a,i) x -> if x mod 2 = 0 then
    (a,i+1)
  else
    ((x,i)::a,i+1))
  ([],0)
  lst
in ret
```