CMSC330 Spring 2023 Quiz 2

Proctoring TA:	Name:	
UID:		
This entire quiz pertains to the OCaml Programming I	anguage	
Problem 1: Basics		[Total 3 pt
Please circle True or False for the following statements: OCaml uses a dynamic type system	True False	
Variables can be overwritten in expressions	True False	
Lists in OCaml are allocated sequentially like Arrays in O	True False	
Problem 2: Typing		[Total 8 pt
For the following questions, provide the type of the given (a) fun f a b -> ((f a) ^ "z") :: b	functions.	[2 pts]
(b) fun x -> fun y -> (x (y + 2)) +. 2.		[2 pts]

For the following questions, write an expression of the following types. All pattern matching must be exhaustive and you cannot use type annotations:

(c) int -> float -> (int * float) list

(d) 'a -> 'b -> 'b -> 'a -> 'a list

[2 pts]

[2 pts]

1

For questions 3 and 4, you may use the following higher order functions:

let rec fold f a l = match l with [] -> a | h::t -> fold f (f a h) t
let rec foldr f l a = match l with[] -> a | h::t -> f h (foldr f t a)
let rec map f l = match l with[] -> [] | h::t -> (f h)::(map f t)

Problem 3: Code Completion

[Total 4 pts]

Write a function called parity_sum that takes in an int list and returns an int * int * int tuple where the first value in the tuple is the sum of even indices, the second is the sum of the odd indices and the third value is the size of the list. Lists are o indexed.

Example:

```
parity_sum [1;2;3;4] = (4,6,4)
parity_sum [1;10;2;20;3;30] = (6,60,6)
parity_sum [] = (0,0,0)
parity_sum [-1;-5;1,5] = (0,0,4)
```

```
let partity_sum lst = fold (fun acc x -> ___Blank_1___ ) ___Blank 2___ arr
```

Blank 1 (3 pts):

Blank 2 (2pts) :

Problem 4: Coding

[Total 5 pts]

Duplicate Create a function dup_elems lst n that duplicates every element in lst n times

Example:

dup_elems [1;2;3;4] 1 = [1;1;2;2;3;3;4;4] dup_elems [] 2 = [] dup_elems ["a";"a";"b";"c"] 0 = ["a";"a";"b";"c"]