Problem 1 (20 points). For each of the following expressions $e$, decide whether a type can be assigned to $e$. If you think $e$ is typable, specify $\tau$ such that $e : \tau$, and also precisely specify $v$ such that $e \Downarrow v$; for both typing and evaluation, informally explain your assertions in English. If a given expression is not typable, briefly explain why. *Note: one of the points of this problem is to practice your understanding, not just report what the OCaml interpreter says. For example, saying that a function evaluates to $<\text{fun}>$ (as the interpreter abbreviates it) is not sufficient. This is why some explanation is required.

a. (fun (x : bool) -> if x then 1.0 else 2.0)
b. (fun (x : bool) -> if x then 1.0 else false)true
c. (5.0, (fun (x : int) -> x = 1))
d. let x : bool = true in (fun (y : int) -> x + y)
e. let x : int = 2 in (fun (y : int) -> x * y)
f. (snd(5.0, (fun (x : int) -> (x, true)))) 10

Problem 2 (20 points). Specify the evaluation result of any of the following expressions that could evaluate meaningfully using static scoping, and likewise for dynamic scoping. For any expression that is typable using static scoping, also specify the scope of each declared variable.

a. let x = 2 in
   let f = (fun (y : int) -> x * y) in
   let x = true in f 5
b. let x = true in
   let f = (fun (y : int) -> x * y) in
   let x = 2 in f 5

Problem 3 (20 points). The Fibonacci sequence is a sequence of numbers defined as follows: the 0th Fibonacci number is 0, the 1st Fibonacci number is 1, and for all $n > 1$, the $n$th Fibonacci number is the $n-1$th plus the $n-2$th Fibonacci number, so we have:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

This sequence turns out to occur quite frequently in nature; among other things, it describes the rate of growth of rabbit reproduction.

For all $n > 0$, we denote the $n$th Fibonacci number as $\text{fib}_n$. Note that we can define the Fibonacci sequence recursively, as follows:

$$\text{fib}_0 = 0$$
$$\text{fib}_1 = 1$$
$$\text{fib}_n = \text{fib}_{n-1} + \text{fib}_{n-2}$$

For this problem, write a recursive function called $\text{fibnum} : \text{int} \rightarrow \text{int}$ with the following specification:

$$\text{For all } n \geq 0, \text{fibnum}(n) \Downarrow \text{"fib}_n\text{"}$$

Also specify which cases in your definition of $\text{fibnum}$ are base cases, which are recursive cases, and which actual parameters in its domain type will cause it to diverge (if any).
Problem 4 (20 points). Identify each declared variable, along with its scope, in your definition of fibnum. You may use informal diagrams, as long as they're clear.

Problem 5 (20 points). A predicate on \( \tau \) is a function with domain type \( \tau \) that returns true or false; that is, it is a function of type \( \tau \rightarrow \text{bool} \). Define predicates on int named is_even and is_odd, where \( \text{is\_even}(n) \downarrow \text{true} \) iff \( n \) is even, and \( \text{is\_odd}(n) \downarrow \text{true} \) iff \( n \) is odd.