Quiz 2 - OCaml

Q1 OCaml Typing
1. For the following two subquestions, you are not allowed to use type annotations.
   - All operations on matching must be exhaustive.
   - No other warnings should be raised.
2. OCaml Typing
   a. Write an OCaml expression of type int → int → bool.
   ```ocaml
   fun x y -> not (x > y)
   ```
3. OCaml Typing
   b. Write an OCaml expression of type (a, b) → a × b.
   ```ocaml
   fun (x, y) -> x × y
   ```

Q2 OCaml Typing2
4. OCaml Typing2
   a. Write the type of the following expression.
   ```ocaml
   let rec f x y = if x > y then f x y else f (y + 1) x
   ```
   (1) int → int → int → int
   (2) int → int → int
   (3) int → int
   ```ocaml
   (1)
   ```

Q3 OCaml Typing2
5. Write the type of the following expression:
   ```ocaml
   let f x = 3 + x
   ```
   a. int → int
   ```ocaml
   (1)
   ```

Q4 OCaml Coding
6. Write a function `sum_list` which returns a sum of all elements of a list and each element of list must be an integer.
7. You can assume that the list will not be empty.
8. You can use the following definitions of `map` and `fold` if you cannot use the `List` module.
9. For example:
   ```ocaml
   let rec sum_list l = match l with
   | [] -> 0
   | x :: xs -> x + sum_list xs
   ```
   ```ocaml
   (1)
   ```