TD9 : Typage `OCaml`

Question 1 : Donner les types, les valeurs et les noms des expressions suivantes :

(a) `2 + 3`
(b) `let x = 3. + 4.`
(c) `let x = 2 in 1 + x`
(d) `let y = let x = 2 in 1 + x`
(e) `let x = 1 in let y = 2 in x < y`
(f) `let x = 2 in let y = -1 + x in 2 * y`
(g) `let somme = let un = "1" in let deux = "2" in un ^ " + " ^ un ^ " = " ^ deux`

Question 2 : Donner les types, les valeurs et les noms des expressions suivantes :

(a) `let x = 2 in let x = 1 in 3 * x`
(b) `let z = let x = 3 in let y = x in let x = 1 in 5 * x + y`
(c) `let y = 2 in let x = y + 1 in let y = x + 3 in x + y`
(d) `let x = 2 in let x = 2 * x in x = 2 * x`
(e) `let x = 2 in let x = 2 * x + let x = 2 * x`
(f) `let x = let x = 2 in let x = 2 * x in let x = 2 * x in x`

Question 3 : Déterminer le type et le comportement des fonctions suivantes :

(a) `let f = fun g -> g 0`
(b) `let f g = g ’a’`
(c) `let f g h = fun x -> (g x)+ (h x)`
(d) `let f g x = g x`
(e) `let f x y z = x y z;`
(f) `let f x y z = (x y)z;`
(g) `let f x y z = x (y z);`
(h) `let comp = fun g f -> fun x -> g (f x)`

Question 4 : Donner des fonctions en `ocaml` ayant les types suivants

(a) `int -> int -> int`
(b) `(int -> int)-> int`
(c) `int -> (int -> int)`