Write the type of each of the following OCaml expressions in the first blank provided, or ill-typed if the expression does not type check. Then, after the ⇒ symbol, write the most simplified value of the expression, or leave it blank if it’s ill-typed.

(a) let a : int list list = [3::[]] ⇒ [[3]]
(b) let b : ill-typed = [(1::2)::[]] ⇒
(c) let c : int list list = [1::[2;3;4]] ⇒ [[1;2;3;4]]
(d) let d : ill-typed = [[1::2];3::4] ⇒ [[1;2;3;4]]
(e) let e : int list list = [[1;2;3;4];[]] ⇒ [[1;2;3;4]]
(f) let f : int list list list = [[1::[]];[2::[3;4];[]]] ⇒ [[[1];[2];[3;4]]]
(g) let g : ill-typed = [(5,12,7);(2,5)] ⇒
(h) let h : int list list = [[5;12;7];[2;5]] ⇒ [[5;12;7];[2;5]]

Simplify the complex expression below, using step by step substitution. What is the value computed for ‘answer’ in the following program? (Note: the @ operator glues two lists of the same type together.)

```
let answer : int list =
  let list = [4;5] in
  let f (num : int) : int list = num :: list in
  let list = [6;7] in
  (f 3) @ list
⇒

let answer : int list =
  let f (num : int) : int list = num :: [4;5] in
  let list = [6;7] in
  (f 3) @ list
⇒

let answer : int list =
  let list = [6;7] in
  (3 :: [4;5]) @ list
⇒

let answer : int list =
  (3 :: [4;5]) @ [6;7]
⇒

let answer : int list =
  [3;4;5] @ [6;7]
⇒

let answer : int list =
  [3;4;5;6;7]
```
Which of the following pieces of code are well-formed OCaml expressions? Refer to the production rules on the back of this sheet. For expressions that do not follow the syntax rules, write *ill-formed*. For those that do, add parentheses and/or underline subexpressions to clarify the boundaries of each expression. The first one had been done for you as an example.

(a) \[\text{if } (x > 1) \text{ then if } (y > 2) \text{ then } 0 \text{ else } 1 \text{ else if } (z > 3) \text{ then } 2 \text{ else } 3\]
    \[\text{if } (x > 1) \text{ then } (\text{if } (y > 2) \text{ then } 0 \text{ else } 1) \text{ else } (\text{if } (z > 3) \text{ then } 2 \text{ else } 3)\]

(b) \[\text{let } x = 5 \text{ in } (\text{if } (x = r) \text{ then } 7 \text{ else } x)\]

(c) \[\text{if } (a = 0) \text{ then } 9 \text{ else if } (a = 1) \text{ then } 8 \text{ else if } (a = 2) \text{ then } 7\]
    *ill-formed (missing else)*

(d) \[\text{if } (a = 0) \text{ then } 9 \text{ else } \text{if } (a = 1) \text{ then } 8 \text{ else } (\text{if } (a = 2) \text{ then } 7 \text{ else } 6)\]

(e) \[\text{if } (a = 0) \text{ then } 9 \text{ else if } (a = 1) \text{ then } 8 \text{ else } 7 \text{ else } 6\]
    *ill-formed (too many else)*

(f) \[\text{if } (x < y) \text{ then } (\text{let } z = y-x \text{ in } z*z) \text{ else } (\text{let } z = x-y \text{ in } z*z)\]

(g) \[\text{let } b = (\text{if } (c = 6) \text{ then } d \text{ else } e) \text{ in } (\text{if } (f = b) \text{ then } g \text{ else } h)\]