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

let add x y = x + y

fold add 0 [1;2;3]  
→ <cl> 0 [1;2;3]  
→ <cl> [1;2;3]  
→ 0

fun a ->
  fun l ->
    match l with
    | [] -> a |
    | (h::t) ->
      fold f (f a h) t

if []
  then
    fold add (add 0 1) [2;3]
  else
    fold add (add 1 2) [3]

→ <cl> 1 [2;3]  
→ fold add (add 1 2) [3]

→ <cl> (0+1) [2;3]  
→ <cl> (0+1) [2;3]  
→ 1

else
  then
    fold add (add 0 1) [2;3]
  else
    fold add (add 1 2) [3]

→ <cl> [2;3]
→ fold add (add 1 2) [3]

→ <cl> (0+1) [2;3]  
→ <cl> (0+1) [2;3]  
→ ...