STUDENTS:
Serge Nasr 260463592
Sorin Muchi 260575810

Q1.1:

- Theorem: reflect (reflect t) = t

- Proof:
    - Base case:
      reflect (reflect Node(x, Empty, Empty))
      -> reflect (Node(x, Empty, Empty))
      -> Node(x, Empty, Empty)

- Induction Hypothesis:
  reflect (Node (x, reflect right, reflect left))
  -> Node(x, left, right)

- Induction Clause:
  reflect (reflect t)
  -> reflect (reflect Node(x, left, right))
  -> reflect (Node (x, reflect right, reflect left))
  -> Node (x, reflect (reflect right), reflect(reflect left))
  -> Node (x, left, right) = t

  Which is what we wanted to prove

Q1.2:

- Theorem: for all m, size m = size’ m 0

- Lemma:
  for all m, acc, size’ m acc = 1 + size’ m (acc - 1) with n being a constant

  Proof of Lemma:
  if m = Empty,
  size’ m acc = acc (by definition of function)
  size’ m (acc-1) = acc - 1 (by definition of function)
  so size’ m acc = 1 + size’ m (acc - 1)

  if m = Node(x, left, right),
  size’ m acc = size' left (size' right (1 + acc))
  size’ m (acc-1) = size' left (size' right (1 + (acc - 1))) = size' left (size' right acc)

Since size’ will return acc then size’ m (1 + acc) = 1 + acc which is equal to 1 + size' left (size' right acc) = 1 + acc

- Base case:
  if m = Empty,
  size m = 0 (by definition of function)
  size’ m 0 = 0 (by definition of function)
  Thus size m = size’ m 0

- Induction Hypothesis:
  for m = Node(x, left, right),
  we assume: size left = size’ left 0 and size right = size’ right 0 (from base case)

- Induction Clause:
  for m = Node(x, left, right),

  size m = 1 + size left + size right (by definition of function)

  size’ m 0 = size' left (size' right 1)
  = size' left (1 + size' right 0) (by Lemma)
  = 1 + size' right 0 + size’ left 0 => 1 + size right + size left (by the induction hypothesis)

  Which is what we wanted to prove