**1.**

**a. Write a predicate to determine the lowest common multiple of a list formed from integer numbers.**

a, b – 2 numbers

counter – a number, initially 1

lcm(a, b, counter) =

* counter, if counter % a = 0 and counter % b = 0
* lcm(a, b, counter + 1), otherwise

l1…ln – the list

lcm\_list(l1…ln) =

* 0, if I = {}
* l1, if n = 1
* lcm(l1, l2, 1), if n = 2
* lcm(l1, lcm\_list(l2…ln), 1), otherwise

**b. Write a predicate to add a value v after 1-st, 2-nd, 4-th, 8-th, ... element in a list.**

x – a number

check – current power of 2, initially 1

check\_power\_2(x, check) =

* false, if x < check
* true, if x = check
* check\_power\_2(x, check \* 2), otherwise

l1…ln – the list

e – the value

counter – a number, initially 1

add\_value(l1…ln, e, counter) =

* {}, if l = {}
* l1 U e U add\_value(l2…ln, e, counter + 1), if check\_power\_2(counter, 1)
* l1 U add\_value(l2…ln, e, counter + 1), otherwise