Assume the loop invariant holds at the beginning of the loop:

r * POW(b,e) == POW(x,y) (LI)

We need to show that

r' * POW(b',e') == POW(x,y) (LI')

where r' b' e' are the values of r, b, and e at the end of one execution of the loop body.

The proof requires different cases for when e is even and when e is odd.

Case where e is even:

In this case,

r' == r

b' == b * b

e' == e/2

Therefore

r' * POW(b',e')

== r * POW(b*b, e/2) by definition of r' b' e'

== r * POW(b,e) by Lemma 1

== POW(x,y) by the fact that the LI was true at the beginning of the loop

Case where e is odd:

In this case,

r' == r * b

b' == b * b

e' == e/2

Therefore

r' * POW(b',e')

== r * b * POW(b*b, e/2) by definition of r' b' e'

== r * POW(b,e) by Lemma 2

== POW(x,y) by LI

Have someone check your work up to this point!

Loop invariant: e >= 1;

Init: The loop invariant holds just before the loop is executed because

initially, e=y, and y≥1 by the precondition

Preservation:

Assume e≥1 (the loop invariant) holds before the loop is run.

To show: e'≥1 [loop invariant is still true at the end]

By definition, e' == e/2

Because the loop test is true, e > 1

Therefore e/2 ≥ 1 (because e≥2, and 2/2=1)

note that the loop test is used, but the fact that the LI is true at the beginning is not

Because we have shown init and preservation, the loop invariant holds at the end of the loop, so we know that, after the loop,

e ≥ 1

Moreover, after the loop, the loop exit test must evaluate to false, so

!(e > 1)

Therefore e == 1 at the end of the function. because if e≥1 and e≠1, then it can only be 1

Have someone check your work up to this point!