Find square root:

Before the nth iteration, the (n-1)^2 == x and loops terminates or it does not equal x and the nth iteration runs.

Initialization:
The loop invariant is correct before the first iteration.

case 1: x == 0, (1-1)^2 == x

case 2: x == some number, and (1-1)^2 != x

Brain kinda stopped here not sure if I can do this lol. Or I have to use words instead.

Maintenance:

Is_prime():
Before the nth iteration x is found to not be a prime number or it is unknown if x is a prime number.

Product by addition:
A bound function for Product by addition(a,b) cand be f(a,b) = a

1, for base case f(a,b) <= 0
2, for the recursive call f(a,b) > 0
3, f(a-1, b) < f(a,b), so product by addition will always terminate.

Claim: product by addition will always return b * a.

Proof:
Let P(a,b) represent the function pba.

Base case:
When a = 0, 0 is returned as required since anything multiplied by 0 = 0.

Inductive hypothesis:
Assume that P(k,b) correctly returns b * k. K > 0.

Inductive claim:
We want to show that P(k+1,b) correctly returns b * k+1.

Inductive proof:
P(k+1, b) returns b + P(k+1-1, b) = b + P(k, b)
P(k,b) returns b * k by IH.

b + b*k = b * (k+1) is returned by P(k+1, b) as needed
Therefore, P(a,b) calculates b * a correctly.

Since P(a,b) always Eg terminates and calculates b*a correctly. It is correct for any integer input. Or w/e it is in python.