Find square root:

Before the nth iteration, the square root has been found or it has not been found.

Wanted to do (n-1)^2 == x for the found case but it does not work with i = 0.

Initialization:
The loop invariant is correct before the 0th iteration since it falls under the case that the square root has not been found since the loop has not yet ran.

Maintenance:
Brained stopped will continue tomorrow.

Is_prime():
Before the nth iteration x is found to not be 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.