1. Prove the correctness of the following algorithm.

```
function MULT(m, n)
    # Precondition: m ∈ ℕ, n ∈ ℤ
    x = m
    y = n
    z = 0
    # Loop Invariant: z = m × n − x × y
    while x ≠ 0 do
        if  x % 2 == 1 then
            z = z + y
        end if
        y = y ≪ 1                         ▷ left shift, equivalent to y = y × 2
        x = x ≫ 1                         ▷ right shift, equivalent to x = ⌊x/2⌋
    end while
    return z
    # Postcondition: returns m × n
end function
```

*Ans:*Recall that the right shift operator $\gg$ removes some number of bits from the right end of its first operand. For example

$$37 \gg 1 = (100101) \gg 1 = (10010) = 18 = \lfloor \frac{37}{2} \rfloor$$
$$26 \gg 2 = (11010) \gg 2 = \underbrace{(1101)}_{13} \gg 1 = (110) = 6$$

Similarly, the left shift operator $\ll$ adds 0s to the right end of its first operand. For example

$$12 \ll 1 = (1100) \ll 1 = (11000) = 24 = 2 \times 12$$
$$13 \ll 2 = (1101) \ll 2 = \underbrace{(11010)}_{26} \ll 1 = (110100) = 52$$

Now, in order to prove the correctness of algorithm, we have to prove for all inputs that satisfy precondition, postcondition holds after execution. But except the few initial assignments and a final return statement, the code implements a loop, so we have to focus on proving the correctness of the loop.

Partial Correctness: We know that proving the correctness of the loop requires us to prove the loop invariant. We have been suggested a loop invariant. Let's see if it is a reasonable choice for proving the correctness.

1. Before the first iteration (iteration 0), we have $x = m$, $y = n$, and $z = 0 = m \times n - x \times y$.
2. Upon loop termination ($x = 0$), we have $z = m \times n - 0 = m \times n$. But that is exactly what we want as the algorithm returns z after the loop.

So let's prove this loop invariant.

Proof. We will prove by induction on the iteration number that $z_i = m \times n - x_i \times y_i$ (loop invariant) in which the subscript i defines the iteration number and v_i is the value of v at the end of iteration i.

Base case: At iteration 0 (before execution of loop), $x_0 = m$, $y_0 = n$, $z_0 = 0$ so

$$z_0 = 0 = m \times n - m \times n = m \times n - x_0 \times y_0$$

Induction step: Let $k \geq 0$ and suppose $z_k = m \times n - x_k \times y_k$ at the end of iteration k (IH). We want to prove $z_{k+1} = m \times n - x_{k+1} \times y_{k+1}$ at the end of iteration $k+1$. Consider two cases:

Case 1: Assume there is no iteration number $k+1$. Then $z_{k+1} = z_k$, $x_{k+1} = x_k$, $y_{k+1} = y_k$. Hence by IH, the loop invariant holds.

Case 2: If there is an iteration number $k+1$. Then by loop condition $x_k \neq 0$. Moreover, $y_{k+1} = 2y_k$ (line 11) and $x_{k+1} = \lfloor \frac{x_k}{2} \rfloor$ (line 12). Now, consider the following cases:

Subcase A: $x_k \;\%\; 2 = 0$ (x_k is even).

$$\begin{aligned}
z_{k+1} &= z_k \\
&= m \times n - x_k \times y_k && \text{(by IH)} \\
&= m \times n - (x_k/2) \times (2y_k) && (x_k \text{ is even}) \\
&= m \times n - x_{k+1} \times y_{k+1} && \text{(line11-12)}
\end{aligned}$$

Subcase B: $x_k \;\%\; 2 = 1$ (x_k is odd).

$$\begin{aligned}
z_{k+1} &= z_k + y_k \\
&= m \times n - x_k \times y_k + y_k && \text{(by IH)} \\
&= m \times n - (x_k - 1) \times y_k \\
&= m \times n - (\tfrac{x_k - 1}{2}) \times (2y_k) && (x_k \text{ is odd}) \\
&= m \times n - x_{k+1} \times y_{k+1} && \text{(line11-12)}
\end{aligned}$$

In all subcases, $z_{k+1} = m \times n - x_{k+1} \times y_{k+1}$. Therefore, by induction, the loop invariant holds for all $i \in \mathbb{N}$. $\square$

Termination: In order to prove the termination, we need to find a decreasing sequence of natural numbers. By the loop condition, the loop terminates at iteration i if $x_i = 0$. Moreover, by line 12, $x_{i+1} = \lfloor \frac{x_i}{2} \rfloor$. For any natural number $s > 0$, it is easy to see that $\lfloor \frac{s}{2} \rfloor \leq \frac{s}{2} < s$. Hence we can conclude that $x_{i+1} < x_i$. By precondition, $x_0 \in \mathbb{N}$ and hence x_i is a decreasing sequence of natural numbers and by theorem 2.5 in the textbook it should be finite, i.e., the loop terminates.

$\square$