```
PS D:\VSC\MATHEMATICS PYTHON\Lab 5>
Enter value for a: 7
Enter value for b: 5
Enter value for m: 13
Solution: x ≡ 10 (mod 13)
```

```python
from math import gcd
# Function to find modular inverse using Extended Euclidean Algorithm
def mod_inverse(a, m):
    if gcd(a, m) != 1:
        return None  # No modular inverse if gcd is not 1
    # Extended Euclidean Algorithm
    m0, x0, x1 = m, 0, 1
    while a > 1:
        q = a // m
        m, a = a % m, m
        x0, x1 = x1 - q * x0, x0
    return x1 + m0 if x1 < 0 else x1  # Ensure positive result
# Function to solve linear congruence equation ax \equiv b (mod m)
```

```python
def solve_linear_congruence(a, b, m):
    g = gcd(a, m)
    # If gcd(a, m) does not divide b, no solution exists
    if b % g != 0:
        return None
    # Reduce the equation by dividing everything by gcd
    a, b, m = a // g, b // g, m // g
    # Find modular inverse of a modulo m
    inv_a = mod_inverse(a, m)
    if inv_a is None:
        return None
    # Compute x as (b * inv_a) % m
    x = (b * inv_a) % m
    return x

# User Input
a = int(input("Enter value for a: "))
b = int(input("Enter value for b: "))
m = int(input("Enter value for m: "))

solution = solve_linear_congruence(a, b, m)
if solution is not None:
    print(f"Solution: x \u2261 {solution} (mod {m})")
else:
    print("No solution exists.")
```