To find one of the dividend, divisor, quotient, or remainder using the slow and fast division algorithm with the non-restoring method, you can follow these steps:

Slow Division Algorithm:

Initialize the dividend (a), divisor (b), quotient (q), and remainder (r) to their given values.
If the dividend (a) is negative, make it positive and set a flag to indicate that the result will be negative. Similarly, if the divisor (b) is negative, make it positive and set the flag accordingly.
Initialize a register (R) with the absolute value of the dividend (|a|).
Initialize a counter (C) to the number of bits in the register (R).
Initialize a working register (W) with the value of the divisor (b).
Repeat the following steps until the counter (C) becomes zero:
a. Shift the register (R) left by one bit, inserting a 0 in the least significant bit.
b. Shift the working register (W) left by one bit, inserting a 0 in the least significant bit.
c. If the working register (W) is less than or equal to the register (R), subtract the working register (W) from the register (R), and set the corresponding bit in the quotient (q) to 1.
d. If the working register (W) is greater than the register (R), set the corresponding bit in the quotient (q) to 0.
e. Decrement the counter (C) by 1.
If the flag indicating a negative result was set initially, negate both the quotient (q) and the remainder (r).
The final quotient (q) and remainder (r) represent the result of the division.
Fast Division Algorithm:

Initialize the dividend (a), divisor (b), quotient (q), and remainder (r) to their given values.
If the dividend (a) is negative, make it positive and set a flag to indicate that the result will be negative. Similarly, if the divisor (b) is negative, make it positive and set the flag accordingly.
Initialize a register (R) with the absolute value of the dividend (|a|).
Initialize a counter (C) to the number of bits in the register (R).
Initialize a working register (W) with the value of the divisor (b).
Repeat the following steps until the counter (C) becomes zero:
a. Shift the register (R) left by one bit, inserting a 0 in the least significant bit.
b. Shift the quotient (q) left by one bit, inserting a 0 in the least significant bit.
c. Subtract the working register (W) from the register (R), and set the corresponding bit in the quotient (q) to 1.
d. If the result of the subtraction is negative, restore the register (R) by adding the working register (W) back to it and set the corresponding bit in the quotient (q) to 0.
e. Decrement the counter (C) by 1.
If the flag indicating a negative result was set initially, negate both the quotient (q) and the remainder (r).
The final quotient (q) and remainder (r) represent the result of the division.
Using the given examples:

If a = 9 and b = 2:

Slow Division Algorithm:

Initialize a = 9, b = 2, q = 4, and r = 1.
Apply the slow division algorithm as described above.
Fast Division Algorithm:

Initialize a = 9, b = 2, q = 4, and r = 1.
Apply the fast division algorithm as described above.
If a = 12 and b = 17:

Slow Division Algorithm:

Initialize a = 12, b = 17, q = 0, and r = 12.
Apply the slow division algorithm as described above.
Fast Division Algorithm:

Initialize a = 12, b = 17, q = 0, and r = 12.
Apply the fast division algorithm as described above.
If a = -17 and b = 3:

Slow Division Algorithm:

Initialize a = -17, b = 3, q = -6, and r = 1.
Apply the slow division algorithm as described above.
Fast Division Algorithm:

Initialize a = -17, b = 3, q = -6, and r = 1