Euclid's algorithm.

If $a = qb + r$ then $\gcd(a, b) = \gcd(b, r)$

because Any common divisor of b, r also divides a. and since, $r = a - qb$, Any common divisor of a, b also divides r.
∴ (a, b) and (b, r) have same common divisor.
so they have same gcd.

Now we use this fact and simplify the calculations of gcd.

$$a = q_1 b + r_1$$
$$b = q_2 r_1 + r_2$$
$$r_1 = q_3 r_2 + r_3$$
$$\vdots$$
$$r_{n-2} = q_n r_{n-1} + r_n$$
$$r_{n-1} = q_{n+1} r_n + 0.$$

From euclids observation we know,

$$\gcd(a, b) = \gcd(b, r_1) = \gcd(r_1, r_2) \ldots = \gcd(r_n, 0)$$

and $\gcd(r_n, 0) = r_n$ --

$$\therefore \boxed{\gcd(a, b) = r_n}$$

this is euclids algorithm.

lets
~~We~~ set $r_0 = a$ and $r_1 = b$.

then we divide r_0, r_1 which gives us reminder r_3

i.e. $r_0 = a$

$r_1 = b$

$r_3 = r_0 - q_1 r_1$ and $0 \leq r_3 < |r_1|$
(this defines q_1)

$\vdots$

$r_{i+1} = r_{i-1} - q_i r_i$ and $0 \leq r_{i+1} < |r_i|$
(defines q_i)

$\vdots$

The computation stops when ~~some~~ reminder ~~r_{k+1}~~ reaches $r_{k+1} = 0$ and last non-zero reminder ~~is~~ r_k is the gcd.

In the code;
we set $x = \max(a,b)$ and $y = \min(a,b)$.
to gcd (x,y).
here $r_0 = x$, $r_1 = y$.
we set $z = x \% z$, ..(remainder function)
it gives us remainde of division x/y.

So. z is r_3. we check if $r_3 = 0$ or not.

then we run while loop.
we store z in variable z_0.
set $x = y$. and $y = z$.
i.e. $x_0 = r_2$ and $y = r_3$.

then we get new z by dividing r_2/r_3 and geting remainder r_4 as z.

this loop keep running untill $z = 0$. that is we reached at $r_{k+1} = 0$. so ther previous remainder the last non zero reminder r_k is gcd. which is stored in z_0.
so z_0 is gcd(a,b).

Euclids extended algorithm.

$\forall\ a, b \in \mathbb{Z}$ (at least one of a, b is not zero), ~~$\exists\ s_0, y_0 \in \mathbb{Z}$~~
~~such that~~
$\exists\ s, t \in \mathbb{Z}$ such that $\gcd(a,b) = as + bt$.

(from Bezouts identity)

we use euclids extended algorithm to find
$\gcd(a,b)$, s, t.
we can express $r_i = as_i + bt_i$.

we set, $r_0 = a$, $r_1 = b$
$s_0 = 1$, $s_1 = 0$
$t_0 = 0$, $t_1 = 1$

($s_0, t_0 = 1, 0$ because $r_0 = a = a\times 1 + b\times 0$)
and ($s_1, t_1 = 0, 1$ because $r_1 = b = a\times 0 + b\times 1$).

from euclids algorithm we know,

$$r_2 = r_0 - q_1 r_1$$
$$\vdots$$
$$r_{i+1} = r_{i-1} - q_i r_i$$

also we can write

$$s_2 = s_0 - q_1 s_1$$
$$\vdots$$
$$s_{i+1} = s_{i-1} - q_i s_i$$
$$\vdots$$

and similarly $t_{i+1} = t_{i-1} - q_i t_i$. — (Proof is skip)

Date:

So, when remainder reaches some $r_{k+1}=0$.
r_k the last non-zero remainder is gcd.
and s_k and t_k is the minimal pair of Bezout coefficient.

So. $\boxed{r_k = as_k + bt_k}$

Now, In my code.

- The code take input a and b.
- check at least one of them is non-zero.
- If the a or b are negative integers then convert them into positive integers. because it makes calculations easy and don't change result. (∵ change in sign do not change gcd).
- set x as max(a,b) any y = min(a,b).
- also checks if one of a, b is zero or not. if one of them is zero then we can directly say that other mod of other no. is gcd.
- and Bezout coefficient is (1,0) (∵ x > y)

- then we set $s_0, t_0, s_1, t_1, r_0, r_1$ accordingly.

- Run while loop.
  Inside loop, we set quotient 'q' by dividing r_0 by r_1 (floor division function).
  Set remainder r by $r_0 \% r_1$ --(gives remainder)
  set s, t according to formulae.

- then we check if $r=0$ or not. if not then we set $r_0 = r_1$, $s_0 = s_1$, $t_0 = t_1$, and $r_1 = r$, $s_1 = s$, $t_1 = t$.

here r, s, t are actually r_2, s_2, t_2. and again loop runs, and r, s, t became r_3, s_3, t_3, and this process continews untill $r = 0$. i.e. we reached at ~~sem~~ $r_{k+1} = 0$. so. r_k is our gcd. and s_k, t_1 are Bezout coefficients.

so. if $r=0$ the privious remainder is r_1 so r_i is gcd(a, b). and. s_1, t_1 are Bezout coefficients.

and finally we get gcd(a,b), and s,t.