```cpp
// mx + ny = gcd(m, n), runtime error for (m, n) = (0, 0)
ll ex_euclid(ll m, ll n, ll &x, ll &y) {
  if (n == 0) { x = 1; y = 0; return m; }
  ll g = ex_euclid(n, m % n, y, x);
  y -= m / n * x;
  return g;
}

// In case when there is range restriction for (x, y)
class Extended_Euclid {
private:
  bool inrange(ll x, ll y, Pll x_rng, Pll y_rng) {
    if (x_rng.fst <= x && x <= x_rng.snd && y_rng.fst <= y && y <= y_rng.snd) return true;
    else return false;
  }
  bool subst_d(ll &x, ll &y, ll d, Pll x_rng, Pll y_rng) {
    ll xc = x, yc = y;
    Loop(k, 3) {
      x = xc + n / g * (d + k - 1);
      y = yc - m / g * (d + k - 1);
      if (inrange(x, y, x_rng, y_rng)) return true;
    }
    return false;
  }
  ll m, n, g, x, y;
public:
  Extended_Euclid(ll m, ll n) {
    this->m = m;
    this->n = n;
    vll q;
    g = gcd(m, n);
    ex_euclid(m, n, x, y);
  }
  bool solve(ll &x, ll &y, ll z, Pll x_rng = { LLONG_MIN, LLONG_MAX }, Pll y_rng = { LLONG_MIN, LLONG_MAX }) {
    if (z % g != 0) return false;
    else {
      ll q = z / g;
      x = this->x * q;
      y = this->y * q;
      if (inrange(x, y, x_rng, y_rng)) return true;
      if (x_rng.fst != LLONG_MIN) {
        ll d = (x_rng.fst - x) / (n / g);
        if (subst_d(x, y, d, x_rng, y_rng)) return true;
      }
      if (x_rng.snd != LLONG_MAX) {
        ll d = (x_rng.snd - x) / (n / g);
        if (subst_d(x, y, d, x_rng, y_rng)) return true;
      }
      if (y_rng.fst != LLONG_MIN) {
        ll d = (y_rng.fst - y) / (m / g);
        if (subst_d(x, y, -d, x_rng, y_rng)) return true;
      }
      if (x_rng.snd != LLONG_MAX) {
        ll d = (y_rng.snd - y) / (m / g);
        if (subst_d(x, y, -d, x_rng, y_rng)) return true;
      }
      return false;
    }
  }
};
```