```
theory Problem-1
  imports Complex-Main
begin
```

0.1 Problem 1

Solve the equation in the integers:

```
theorem problem1:
  fixes x y :: int
  assumes x ≠ 0 and y ≠ 0
  shows 1 / x² + 1 / (x∗y) + 1 / y² = 1
    ⟷ x = 1 ∧ y = −1 ∨ x = −1 ∧ y = 1
    (is ?eqn ⟷ ?sols)
proof
  — Unfortunately, removing the conversions between int and real takes a few lines
  let ?x = real-of-int x and ?y = real-of-int y
  assume ?eqn
  then have 1 / ?x² + 1 / (?x∗?y) + 1 / ?y² = 1 by auto
  hence ?x²∗?y² / ?x² + ?x²∗?y² / (?x∗?y) + ?x²∗?y² / ?y² = ?x²∗?y²
    by algebra
  hence ?x² + ?x∗?y + ?y² = ?x² ∗ ?y² using ‹x ≠ 0› ‹y ≠ 0›
    by (simp add: power2-eq-square)
  hence inteq: x² + x∗y + y² = x² ∗ y²
    using of-int-eq-iff by fastforce

  define g where g = gcd x y
  then have g ≠ 0 and g > 0 using ‹x ≠ 0› ‹y ≠ 0› by auto
  define x' y' where x' = x div g and y' = y div g
  then have x' ∗ g = x and y' ∗ g = y using g-def by auto
  from inteq and this have g² ∗ (x'² + x' ∗ y' + y'²) = x'² ∗ y'² ∗ g^4
    by algebra
  hence reduced: x'² + x' ∗ y' + y'² = x'² ∗ y'² ∗ g² using ‹g ≠ 0› by algebra

  hence x' dvd y'² and y' dvd x'²
    by algebra+
  moreover have coprime x' (y'²) coprime (x'²) y'
    unfolding x'-def y'-def g-def
    using assms div-gcd-coprime by auto
  ultimately have is-unit x' is-unit y'
    unfolding coprime-def by auto
  hence abs1: |x'| = 1 ∧ |y'| = 1 using assms by auto
  then consider (same-sign) x' = y' | (diff-sign) x' = −y' by fastforce
  thus ?sols
  proof cases
    case same-sign
    then have x' ∗ y' = 1
      using abs1 and zmult-eq-1-iff by fastforce
    hence g² = 3
      using abs1 same-sign and reduced by algebra
    hence 1² < g² and g² < 2² by auto
    hence 1 < g and g < 2
      using ‹g > 0› and power2-less-imp-less by fastforce+
    hence False by auto
    thus ?sols by auto
  next
    case diff-sign
    then have x' ∗ y' = −1
      using abs1
      by (smt mult-cancel-left2 mult-cancel-right2)
    hence g² = 1
      using abs1 diff-sign and reduced by algebra
    hence g = 1 using ‹g > 0›
```

by (*smt power2-eq-1-iff*)
hence $x = x'$ **and** $y = y'$
unfolding x'-*def* **and** y'-*def* **by** *auto*
thus *?sols* **using** *abs1* **and** *diff-sign* **by** *auto*
qed
next
assume *?sols*
then show *?eqn* **by** *auto*
qed

end