1 Series I (September)

1.1 Problem 1

```
theory SeriesI
  imports
    Complex-Main
    HOL−Analysis.Analysis
begin
```

Let a, b be real numbers. Let's assume that, for all real numbers x, y the inequality $|(ax+by)(ay+bx)| \leq x^2+y^2$ is satisfied. Show that $a^2+b^2 \leq 2$.

```
theorem problem1:
  fixes a b :: real
  assumes given: ⋀x y :: real. |(a∗x + b∗y)∗(a∗y + b∗x)| ≤ x² + y²
  shows a² + b² ≤ 2
proof −
  from given [where x=1 and y=1] have (a+b)² ≤ 2
    by (simp add: power2-eq-square)
  moreover from given [where x=1 and y=−1] have (a−b)² ≤ 2
    by (simp add: power2-eq-square right-diff-distrib′)
  ultimately have (a+b)² + (a−b)² ≤ 4 by auto
  moreover have (a+b)² + (a−b)² = 2∗(a² + b²) by algebra
  ultimately show a² + b² ≤ 2 by auto
qed
```

1.2 Problem 3

Let's assume that a positive integer n has no divisor d that satisfies $\sqrt{n} \leq d \leq \sqrt[3]{n^2}$. Prove that n has a prime divisor $p > \sqrt[3]{n^2}$.

```
theorem problem3:
  fixes n :: nat
  assumes [iff]: n ≠ 0
  assumes divrange: ⋀d :: nat. sqrt n ≤ d ⟹ d ≤ n powr (2/3) ⟹ ¬d dvd n
  obtains p where prime p and p > n powr (2/3)
proof −
   have forbidden-range: ¬d dvd n if n powr (1/3) ≤ d and d ≤ n powr (2/3)
for d :: nat
  proof
    assume d dvd n
    from that consider
    (low) n powr (1/3) ≤ d d ≤ sqrt n |
    (high) sqrt n ≤ d d ≤ n powr (2/3)
      by fastforce
    then show False
    proof cases
      case low
      from ‹d dvd n› have mirror-divisor: (n div d) dvd n by auto
```

```
    have n/d ≤ n / n powr (1/3)
      using low by (simp add: frac-le)
    also have ... = n powr 1 / n powr (1/3) by auto
    also have ... = n powr (2/3) by (simp del: powr-one flip: powr-diff)
    finally have n/d ≤ n powr (2/3).
    moreover from ‹d dvd n› have n/d = n div d by auto
    ultimately have upper-bound: n div d ≤ n powr (2/3) by auto

    from ‹d dvd n› have d ≠ 0
      by (meson ‹n ≠ 0› dvd-0-left)
    hence n/d ≥ n / sqrt n
      using low by (simp add: frac-le)
    also have n / sqrt n = sqrt n
      using real-div-sqrt ‹n ≠ 0› by auto
    finally have n/d ≥ sqrt n.
    hence lower-bound: n div d ≥ sqrt n using ‹n/d = n div d› by auto

    show False using divrange [of n div d] mirror-divisor
      and lower-bound upper-bound by auto
  next
    case high
    then show False using divrange ‹d dvd n› by auto
  qed
qed

have n > 1
proof −
  {
    assume n = 1
    with divrange [of 1] have ¬ 1 dvd 1 by auto
    moreover have 1 dvd (1::nat) by auto
    ultimately have False by contradiction
  }
  thus n > 1 using ‹n ≠ 0›
    by fastforce
qed

let ?smalldivs = {d. d dvd n ∧ d < n powr (1/3)}
have finite ?smalldivs using finite-divisors-nat by fastforce
moreover have ?smalldivs ≠ {} proof −
  have 1 ∈ ?smalldivs using ‹n > 1› by auto
  thus ?thesis by auto
qed

moreover define a where a = Max ?smalldivs
ultimately have a ∈ ?smalldivs using Max-in by auto
hence a < n powr (1/3) and a dvd n by auto
hence a ≠ 0 using ‹n ≠ 0› by algebra
have ⋀d. d dvd n ⟹ d > a ⟹ d ≥ n powr (1/3)
```

```
    using Max-ge ‹finite ?smalldivs› ‹?smalldivs ≠ {}› a-def
    by (metis (no-types, lifting) mem-Collect-eq not-le)
  hence div-above-a: ⋀d. d dvd n ⟹ d > a ⟹ d > n powr (2/3)
    using forbidden-range
    by force

  note ‹a < n powr (1/3)›
  also have n powr (1/3) < n powr 1 using ‹n > 1› by (intro powr-less-mono)
auto
  finally have a < n by auto
  hence n div a > 1
    using ‹a dvd n› by fastforce
  then obtain p where prime p and p dvd (n div a)
    by (metis less-irrefl prime-factor-nat)
  hence p*a dvd n using ‹a dvd n› and ‹n div a > 1›
    by (metis div-by-0 dvd-div-iff-mult gr-implies-not-zero)
  with div-above-a [of p*a] have p*a > n powr (2/3)
    using ‹prime p› and prime-nat-iff by fastforce
  moreover have a * n powr (1/3) < n powr (1/3) * n powr (1/3)
    using ‹a < n powr (1/3)› by auto
  moreover have ... = n powr (2/3) by (simp flip: powr-add)
  ultimately have p*a > a*n powr (1/3) by simp
  hence p > n powr (1/3) using ‹a ≠ 0› by simp
  hence p > n powr (2/3) using forbidden-range [of p] and ‹p * a dvd n› by
force
  moreover note ‹prime p›
  ultimately show ?thesis using that [of p] by auto
qed
end
```