

# Project Numina

Assistant Mathématique de Preuve Interactive

---

## Rapport de Trace de Preuve Formelle Lean 4

*Document interactif compilé pour Denise Vella-Chemla*

---

### 1 Énoncé du Problème

#### Énoncé Informel

Prove that for all  $n \in \mathbb{N}$ ,  $n \geq 6$  and  $2 \mid n$ , there exists  $p \in \mathbb{N}$ ,  $3 \leq p \leq n/2$ , such that for all  $q \in \mathbb{N}$  with  $2 \leq q \leq \sqrt{n}$ ,  $p$  is not congruent to 0 (mod  $q$ ) and  $p$  is not congruent to  $n$  (mod  $q$ ).

#### Formalisation Initiale Lean 4

```
import Mathlib

-- Formalization task
-- Write Lean code below

theorem existence_of_special_number :
  ∀ n : ℕ, n ≥ 6 → Even n →
    ∃ p : ℕ, 3 ≤ p ∧ p ≤ n / 2 ∧
      ∀ q : ℕ, 2 ≤ q → q ≤ Nat.sqrt n →
        ¬(q ∣ p) ∧ ¬(p ≡ n [MOD q]) := by
  sorry
```

## 2 Trace des Étapes et Réflexions

### Étape 1 : Réflexion de l'Agent (Thinking)

#### Pensée cognitive interne

I'll help you complete the proofs in the Lean file. Let me start by reading the files to understand the problem.

### Étape 2 : Consultation de l'état (mcp \_\_ lean-lsp \_\_ lean \_goal)

#### Serveur Lean 4 - État actuel du but à prouver

Buts actifs sur la ligne 11 :

```
⊢ ∀ n ≥ 6, Even n → ∃ p, 3 ≤ p ∧ p ≤ n / 2 ∧
  ∀ (q : ℕ), 2 ≤ q → q ≤ n.sqrt → ¬q ∣ p ∧ ¬p ≡ n [MOD q]
```

### Étape 3 : Réflexion cognitive (Thinking)

#### Analyse mathématique

This is a challenging number theory problem. The statement claims that for even numbers  $n \geq 6$ , there exists a special  $p$  between 3 and  $n/2$  such that for all  $q$  between 2 and  $\sqrt{n}$ ,  $q$  doesn't divide  $p$  and  $p$  is not congruent to  $n$  modulo  $q$ .

Let me think about this : for an even  $n \geq 6$ , we need to find  $p$  where :

- $3 \leq p \leq n/2$
- For all  $2 \leq q \leq \sqrt{n}$  :  $q$  doesn't divide  $p$  AND  $p \not\equiv n \pmod{q}$

One approach is to try  $p = n/2 - 1$ . Since  $n$  is even,  $n/2$  is an integer. Let's check if  $p = n/2 - 1$  works. Let's first test the small cases  $n = 6$  and  $n = 8$ , then do the general case for  $n \geq 10$  using  $p = n/2 - 1$ .

## Étape 4 : Édition du code source (Edit)

### Outil : Modification du fichier Lean

Modification appliquée :

```
theorem existence_of_special_number :
  ∀ n : ℕ, n ≥ 6 → Even n →
    ∃ p : ℕ, 3 ≤ p ∧ p ≤ n / 2 ∧
      ∀ q : ℕ, 2 ≤ q → q ≤ Nat.sqrt n →
        ¬(q | p) ∧ ¬(p ≡ n [MOD q]) := by
  intro n hn heven
  -- Check small cases first
  by_cases h6 : n = 6
    subst h6
    use 3
    norm_num
    intro q hq_ge hq_le
    have : q = 2 := by omega
    subst this
    norm_num [Nat.ModEq]
  by_cases h8 : n = 8
    subst h8
    use 3
    norm_num
    intro q hq_ge hq_le
    have : q = 2 := by omega
    subst this
    norm_num [Nat.ModEq]
  -- For n >= 10, use p = 5
  have hn_ge : n ≥ 10 := by omega
  use 5
  refine by norm_num, ?_, ? _
    -- 5 <= n / 2
    omega
    -- For all q in range
    intro q hq_ge hq_le
    sorry
```

## Étape 5 : Diagnostic de compilation (mcp\_\_lean-lsp\_\_lean\_\_diagnostic\_\_messages)

### Serveur Lean 4 - Messages d'erreur et avertissements

```
131c29-131c34, severity: 1 (Warning)
declaration uses 'sorry'
```

Le compilateur valide que le code Lean écrit est syntaxiquement et sémantiquement correct, à l'exception de la conjecture intermédiaire encore marquée par `sorry`.

— Fin de la trace de preuve —