

# Project Numina

Assistant Informatique de Preuve Interactive

---

## Trace d'Exécution et Résolution Formelle Lean 4

Généré automatiquement à partir des journaux de l'agent

---

### 1 Énoncé Mathématique

#### Énoncé Informel (Langue Naturelle)

$\forall n, (n \geq 6 \wedge \text{Even } n) \rightarrow (\exists p, 3 \leq p \wedge p \leq n/2 \wedge (\forall q, (3 \leq q \wedge q \leq \sqrt{n}) \rightarrow (\neg(p \equiv 0 \pmod{q}) \wedge \neg(p \equiv n \pmod{q}))))$

#### Énoncé Formel (Lean 4) - Statut : complete

```
import Mathlib

-- Formalization task
-- Write Lean code below

theorem even_number_property (n : ℕ) (h1 : n ≥ 6) (h2 : Even n) :
  ∃ p : ℕ, 3 ≤ p ∧ p ≤ n / 2 ∧
  ∀ q : ℕ, 3 ≤ q ∧ q ≤ Nat.sqrt n →
    ¬(p % q = 0) ∧ ¬(p % q = n % q) := by
  sorry
```

## 2 Journal des Étapes de Preuve

### Étape 1 : Thinking (Réflexion)

#### Pensée de l'Agent Numina

I'll help you prove all the sorries in the Lean file. Let me start by reading the files to understand the problem and current proof state.

### Étape 2 : Appel d'Outil - Read

#### Interaction Système : Read

```
# Problem Description
```

```
∀ n, (n ≥ 6 ∧ Even n) → (∃ p, 3 ≤ p ∧ p ≤ n/2 ∧ (∀ q, (3 ≤ q ∧ q ≤ √n) → ¬(p ≡ 0 mod q) ∧ ¬(p ≡ n mod q))))
```

### Étape 3 : Appel d'Outil - Read

#### Interaction Système : Read

```
import Mathlib

-- Formalization task
-- Write Lean code below

theorem even_number_property (n : ℕ) (h1 : n ≥ 6) (h2 : Even n) :
  ∃ p : ℕ, 3 ≤ p ∧ p ≤ n / 2 ∧
  ∀ q : ℕ, 3 ≤ q ∧ q ≤ Nat.sqrt n →
    ¬(p % q = 0) ∧ ¬(p % q = n % q) := by
  sorry
```

### Étape 4 : Thinking (Réflexion)

#### Pensée de l'Agent Numina

Now let me check the current proof state and diagnostics:

### Étape 5 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

#### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  sorry\nBefore:\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\n¬ ∃ p, 3 ≤ p ∧ p ≤ n / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ n.sqrt → ¬p % q = 0 ∧ ¬p % q = n % q\nAfter:\nno goals"}
```

## Étape 6 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["16c9-16c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 7 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Now I understand the problem. I need to prove that for any even number  $n \geq 6$ , there exists a number  $p$  where:

- $3 \leq p \leq n/2$
- For all  $q$  where  $3 \leq q \leq \sqrt{n}$ :  $p$  is not divisible by  $q$ , and  $p$  and  $n$  have different remainders modulo  $q$

Let me analyze this problem. For even  $n \geq 6$ , I can try  $p = 3$ . Let me verify the conditions:

- $3 \leq 3$  ✓
- $3 \leq n/2$  (since  $n \geq 6$ ) ✓
- For any  $q$  with  $3 \leq q \leq \sqrt{n}$ :  $3 \% q \neq 0$  (need  $q > 3$  for divisibility) and  $3 \% q \neq n \% q$

Let me construct the proof:

## Étape 8 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 9 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["16c9-16c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 10 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  · omega\nBefore:\ncase h.right.left\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nh-  
3 ≤ n / 2\n---\ncase h.right.right\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nh- ∀ (q : ℕ), 3 ≤ q ∧ q ≤  
n.sqrt  
→ ¬3 % q = 0 ∧ ¬3 % q = n % q\nAfter:\nno goals"}
```

## Étape 11 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  constructor\nBefore:\ncase h.right.right\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nq : ℕ\nhq_ge : 3 ≤ q\nhq_le : q ≤ n.sqrt\n¬3 % q = 0 ∧ ¬3 % q = n % q\nAfter:\ncase h.right.right.left\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nq : ℕ\nhq_ge : 3 ≤ q\nhq_le : q ≤ n.sqrt\n¬3 % q = 0\n---\ncase h.right.right.right\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nq : ℕ\nhq_ge : 3 ≤ q\nhq_le : q ≤ n.sqrt\n¬3 % q = n % q"}}
```

## Étape 12 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  · intro h_div\nBefore:\ncase h.right.right.left\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nq : ℕ\nhq_ge : 3 ≤ q\nhq_le : q ≤ n.sqrt\n¬3 % q = 0\n---\ncase h.right.right.right\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nq : ℕ\nhq_ge : 3 ≤ q\nhq_le : q ≤ n.sqrt\n¬3 % q = n % q\nAfter:\ncase h.right.right.left\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nq : ℕ\nhq_ge : 3 ≤ q\nhq_le : q ≤ n.sqrt\n¬3 % q = 0\n¬ False"}}
```

## Étape 13 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Good! Now I need to prove that  $3 \% q \neq 0$  for  $q \geq 3$ . If  $3 \% q = 0$ , then  $q$  divides 3, but the only divisors of 3 are 1 and 3. Since  $q \geq 3$  and  $q$  divides 3, we must have  $q = 3$ . But  $3 \% 3 = 0$ , so this doesn't work.

Let me reconsider. Maybe I should use a different value for  $p$ . Let me try  $p = n/2 - 1$ :

## Étape 14 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 15 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  use n / 2 - 1\nBefore:\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\n¬ ∃ p, 3 ≤ p ∧
```

```

p ≤ n / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ n.sqrt → ¬p % q = 0 ∧ ¬p % q = n % q\nAfter:\ncase h\nn :
ℕ\nh1
: n ≥ 6\nh2 : Even n\nh- 3 ≤ n / 2 - 1 ∧ n / 2 - 1 ≤ n / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ n.sqrt → ¬(n
/ 2
- 1) % q = 0 ∧ ¬(n / 2 - 1) % q = n % q"}

```

## Étape 16 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me think about this problem differently. The statement is asking for a property that seems very specific. Let me search for similar lemmas or theorems:

## Étape 17 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_leandex

### Réponse du Serveur Lean

```
{"result":"leandex error:\n'primary_declaration'"}
```

## Étape 18 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me reconsider the problem. Looking at the statement more carefully, for an even  $n \geq 6$ , we need to find  $p$  such that for all  $q$  in range  $[3, \sqrt{n}]$ ,  $p$  is not divisible by  $q$  AND  $p$  and  $n$  have different remainders mod  $q$ .

Since  $n$  is even,  $n \% 2 = 0$ . If I choose  $p$  to be odd, then  $p \% 2 \neq 0 = n \% 2$ . But we need  $q \geq 3$ .

Let me try a simple candidate:  $p = 5$  for most cases, or adjust based on  $n$ . Actually, let me try to understand what property this is expressing by testing with a concrete example.

For  $n = 6$ :  $n/2 = 3$ ,  $\sqrt{n} \approx 2.45$ , so no  $q$  satisfies  $3 \leq q \leq \sqrt{n}$ . The condition is vacuously true for any  $p$  with  $3 \leq p \leq 3$ , so  $p = 3$  works.

For  $n = 8$ :  $n/2 = 4$ ,  $\sqrt{n} \approx 2.83$ , so still no  $q$  satisfies  $3 \leq q \leq \sqrt{n}$ . Any  $p$  with  $3 \leq p \leq 4$  works.

For  $n = 10$ :  $n/2 = 5$ ,  $\sqrt{n} \approx 3.16$ , so  $q = 3$  is the only value. We need  $p$  such that  $p \% 3 \neq 0$  and  $p \% 3 \neq 10 \% 3 = 1$ . So  $p \% 3$  must be 2. Candidates:  $p = 5$  works ( $5 \% 3 = 2$ ).

Actually, maybe I should just try to prove it exists by casework or by choosing  $p = n/2 - 1$  when  $n \geq 8$ , and  $p = 3$  when  $n = 6$ :

## Étape 19 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 20 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l11c3-l14c13, severity: 1\nunsolved goals\ncase h\nh1 : 6 ≥ 6\nh2 : Even 6\n¬ ∀ (q :  
: N), 3 ≤ q → q ≤ Nat.sqrt.iter 6 (1 <<< (Nat.log2 6 / 2 + 1)) → ¬3 % q = 0 ∧ ¬3 % q = 6 %  
q","l15c32-l15c37, severity: 1\nomega could not prove the goal:\nNo usable constraints found.  
You  
may need to unfold definitions so `omega` can see linear arithmetic facts about `Nat` and  
`Int`,  
which may also involve multiplication, division, and modular remainder by constants."]}
```

## Étape 21 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n · subst hn\nBefore:\ncase pos\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nhn : n  
= 6\n¬ ∃ p, 3 ≤ p ∧ p ≤ n / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ n.sqrt → ¬p % q = 0 ∧ ¬p % q = n %  
q\n---\ncase neg\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nhn : ¬n = 6\n¬ ∃ p, 3 ≤ p ∧ p ≤ n / 2 ∧ ∀ (q :  
: N), 3 ≤ q ∧ q ≤ n.sqrt → ¬p % q = 0 ∧ ¬p % q = n % q\nAfter:\ncase pos\nh1 : 6 ≥ 6\nh2 : Even  
6\n¬  
∃ p, 3 ≤ p ∧ p ≤ 6 / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ Nat.sqrt 6 → ¬p % q = 0 ∧ ¬p % q = 6 % q"}
```

## Étape 22 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me fix the n = 6 case:

## Étape 23 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 24 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l11c3-l15c22, severity: 1\nunsolved goals\ncase h\nh1 : 6 ≥ 6\nh2 : Even 6\nq :  
N\nhq_ge : 3 ≤ q\nhq_le : q ≤ 2\n⊢ -3 % q = 0 ∧ -3 % q = 6 % q"]}
```

## Étape 25 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  intro q hq_ge hq_le\nBefore:\ncase h\nh1 : 6 ≥ 6\nh2 : Even 6\n⊢  
(q : N), 3 ≤ q → q ≤ 2 → -3 % q = 0 ∧ -3 % q = 6 % q\nAfter:\ncase h\nh1 : 6 ≥ 6\nh2 : Even  
6\nq :  
N\nhq_ge : 3 ≤ q\nhq_le : q ≤ 2\n⊢ -3 % q = 0 ∧ -3 % q = 6 % q"}
```

## Étape 26 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Great! We have  $3 \leq q \leq 2$ , which is a contradiction. Let me finish this:

## Étape 27 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 28 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l6c9-l6c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 29 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Good! Now I need to handle the case when  $n \geq 8$ . Let me check the goal:

## Étape 30 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n      omega\nBefore:\nk : ℕ\nh1 : k + k ≥ 6\nhn : ¬k + k = 6\n¬ k + k ≥ 8\nAfter:\nno goals"}
```

## Étape 31 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n      sorry\nBefore:\ncase neg\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nhn : ¬n = 6\nhn_ge_8 : n ≥ 8\n¬ ∃ p, 3 ≤ p ∧ p ≤ n / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ n.sqrt → ¬p % q = 0 ∧ ¬p % q = n % q\nAfter:\nno goals"}
```

## Étape 32 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Now let me think about the case  $n \geq 8$ . I need to find  $p$  such that:

- $3 \leq p \leq n/2$
- For all  $q$  with  $3 \leq q \leq \sqrt{n}$ :  $p \% q \neq 0$  and  $p \% q \neq n \% q$

Since the problem statement seems to claim this always exists, let me try to use a simple choice.

Let me try  $p = n/2 - 1$ :

For  $n = 8$ :  $p = 3$ ,  $\sqrt{8} \approx 2.83$ , so no  $q$  satisfies  $3 \leq q \leq \sqrt{8}$ .

For  $n = 10$ :  $p = 4$ ,  $\sqrt{10} \approx 3.16$ , so  $q = 3$ . We need  $4 \% 3 \neq 0$  (true,  $1 \neq 0$ ) and  $4 \% 3 \neq 10 \% 3$  (true,  $1 \neq 1$ ) - wait that's false!

Let me recalculate:  $10 \% 3 = 1$ ,  $4 \% 3 = 1$ , so they're equal. This won't work.

Let me try  $p = n/2 - 2$  for  $n = 10$ :  $p = 3$ ,  $3 \% 3 = 0$ , which violates the first condition.

Let me try  $p = n/2 + 1$ ... wait, that violates  $p \leq n/2$ .

Hmm, maybe I need a different approach. Let me think about this more carefully. For  $n = 10$ , I need

$p$  such that  $p \% 3 \neq 0$  and  $p \% 3 \neq 1$ . So  $p \% 3 = 2$ . Candidates:  $p = 5$  works! And  $5 \leq n/2 = 5$ .

So perhaps the answer is to choose  $p$  based on the residues. But this is getting complex. Let me try a different strategy - maybe there's a pattern based on whether  $n/2$  is odd or even:

## Étape 33 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 34 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l25c23-l25c28, severity: 1\nTactic `introN` failed: There are no additional binders or `let` bindings in the goal to introduce\n\ncase h.right.right\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\nhn : -n = 6\nhn_ge_8 : n ≥ 8\nnh_sqrt : n.sqrt < 3\nnq : ℕ\nhq_ge : 3 ≤ q ∧ q ≤ n.sqrt\n-3 % q = 0 ∧ -3 % q = n % q"]}
```

## Étape 35 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 36 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l6c9-l6c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 37 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Good! Now for the case when  $\sqrt{n} \geq 3$ , which means  $n \geq 9$ . Let me think about what value of  $p$  to use.

Let me look at specific examples:

-  $n = 10$ :  $\sqrt{n} \approx 3.16$ , so  $q = 3$ .  $n \% 3 = 1$ . Need  $p \% 3 \neq 0$  and  $p \% 3 \neq 1$ , so  $p \% 3 = 2$ .  $p = 5$  works

( $5 \% 3 = 2$ ).

-  $n = 12$ :  $\sqrt{n} \approx 3.46$ , so  $q = 3$ .  $n \% 3 = 0$ . Need  $p \% 3 \neq 0$  and  $p \% 3 \neq 0$ , so  $p \% 3 \in \{1, 2\}$ .  $p = 4$  or

$p = 5$  work.

-  $n = 14$ :  $\sqrt{n} \approx 3.74$ , so  $q = 3$ .  $n \% 3 = 2$ . Need  $p \% 3 \neq 0$  and  $p \% 3 \neq 2$ , so  $p \% 3 = 1$ .  $p = 4$  works.

I notice a pattern: if  $n \% 3 = 0$ , choose  $p$  with  $p \% 3 = 1$ . If  $n \% 3 = 1$ , choose  $p$  with  $p \% 3 = 2$ .

If  $n \% 3 = 2$ , choose  $p$  with  $p \% 3 = 1$ .

But I also need to handle larger  $q$  values. Let me think about the general strategy. Actually, looking at the problem more carefully, maybe I should try using  $n/2 - 1$  or similar values and verify the properties.

Let me try a computational approach for specific small values:

## Étape 38 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 39 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l29c39-l29c53, severity: 1\nType mismatch\n Nat.sqrt_le' n\nhas type\n n.sqrt ^ 2 ≤  
n\nbut is expected to have type\n n.sqrt * n.sqrt ≤ n","l30c9-l30c14, severity: 1\nomega could not  
prove the goal:\na possible counterexample may satisfy the constraints\n 0 ≤ b ≤ 8\n a ≥ 3\nwhere\n a := ↑n.sqrt\n b := ↑n.sqrt * ↑n.sqrt","l31c7-l31c23, severity: 1\ninterval_cases failed: could not find upper bound on n"]}
```

## Étape 40 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me reconsider. The problem statement is asking for an existence proof, not a constructive one necessarily. But in Lean, we need to be constructive. Let me try a different approach - perhaps the statement is only true for specific values or there's a pattern I'm missing.

Let me re-read the natural language description: " $\forall n, (n \geq 6 \wedge \text{Even } n) \rightarrow (\exists p, 3 \leq p \wedge p \leq n/2 \wedge (\forall q, (3 \leq q \wedge q \leq \sqrt{n}) \rightarrow \neg(p \equiv 0 \pmod{q}) \wedge \neg(p \equiv n \pmod{q})))$ "

Wait, let me reconsider the interpretation. This seems like it might be related to Goldbach's conjecture or some number theory result. Let me try a simpler approach: just use  $p = n/2 - 1$  and prove the properties:

## Étape 41 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 42 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l6c9-l6c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 43 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Good! Now let me handle  $n \geq 10$ . Let me look at a pattern for choices:

## Étape 44 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 45 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n      omega\nBefore:\ncase h\nh1 : 8 ≥ 6\nh2 : Even 8\nhn : ¬8 = 6\nq : \n\nh3 : 3 ≤ q\nh4 : q ≤ 2\n¬3 % q = 0 ∧ ¬3 % q = 8 % q\nAfter:\nno goals"}
```

## Étape 46 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me check the final goal:

## Étape 47 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n  use if (2 * k) % 3 = 0 then 4 else if (2 * k) % 3 = 1 then 5 else 4\nBefore:\ncase neg\nk : ℕ\nh1 : k + k ≥ 6\nhn : ¬k + k = 6\nh8 : ¬k + k = 8\nhn_ge_10 : k + k ≥ 10\nhk : k ≥ 5\n¬ ∃ p, 3 ≤ p ∧ p ≤ (k + k) / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ (k + k).sqrt → ¬p % q = 0 ∧ ¬p % q = (k + k) % q\nAfter:\ncase h\nk : ℕ\nh1 : k + k ≥ 6\nhn : ¬k + k = 6\nh8 : ¬k + k = 8\nhn_ge_10 : k + k ≥ 10\nhk : k ≥ 5\n¬ (3 ≤ if 2 * k % 3 = 0 then 4 else if 2 * k % 3 = 1 then 5 else 4) ∧ \n  (if 2 * k % 3 = 0 then 4 else if 2 * k % 3 = 1 then 5 else 4) ≤ (k + k) / 2 ∧ \n  ∀ (q : ℕ), \n    3 ≤ q ∧ q ≤ (k + k).sqrt → \n      ¬(if 2 * k % 3 = 0 then 4 else if 2 * k % 3 = 1 then 5 else 4) % q = 0 ∧ \n      ¬(if 2 * k % 3 = 0 then 4 else if 2 * k % 3 = 1 then 5 else 4) % q = (k + k) % q"}
```

## Étape 48 : Thinking (Réflexion)

### Pensée de l'Agent Numina

This is getting complex. Let me try a simpler approach. Looking at the problem again, maybe I should just try to find a general formula that works. Let me think...

Actually, let me reconsider the problem statement. Maybe it's easier to understand if I verify it for small values manually and then generalize. Let me try using classical logic and assuming such a  $p$  exists:

Actually, wait. Let me reconsider whether this theorem is even true as stated. Let me check  $n = 10$  more carefully:

- $n = 10$ ,  $n/2 = 5$ ,  $\sqrt{10} \approx 3.16$
- $q = 3$ : We need  $p \% 3 \neq 0$  and  $p \% 3 \neq 10 \% 3 = 1$   
So  $p \% 3 = 2$ . Candidates in  $[3, 5]$ :  $p = 5$  ( $5 \% 3 = 2$ ) ✓

Let me check  $n = 12$ :

- $n = 12$ ,  $n/2 = 6$ ,  $\sqrt{12} \approx 3.46$
- $q = 3$ : We need  $p \% 3 \neq 0$  and  $p \% 3 \neq 12 \% 3 = 0$   
So  $p \% 3 \in \{1, 2\}$ . Candidates: 4, 5 work.

Let me check  $n = 16$ :

- $n = 16$ ,  $n/2 = 8$ ,  $\sqrt{16} = 4$
- $q = 3$ : We need  $p \% 3 \neq 0$  and  $p \% 3 \neq 16 \% 3 = 1$ . So  $p \% 3 = 2$ . Candidates: 5, 8.
- $q = 4$ : We need  $p \% 4 \neq 0$  and  $p \% 4 \neq 16 \% 4 = 0$ . So  $p \% 4 \in \{1, 2, 3\}$ .  
From  $p \in \{5, 8\}$ :  $5 \% 4 = 1$  ✓,  $8 \% 4 = 0$  ✗  
So  $p = 5$  works.

It seems like  $p = 5$  is often a good choice. Let me try that:

## Étape 49 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 50 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l26c3-l28c24, severity: 1\nunsolved goals\ncase h\nh1 : 10 ≥ 6\nh2 : Even 10\nhn :  
-10  
= 6\nhn8 : -10 = 8\nhn_ge_10 : 10 ≥ 10\nh- ∀ (q : ℕ), 3 ≤ q → q ≤ 3 → -5 % q = 0 ∧ -5 % q = 10 %  
q","l30c3-l32c24, severity: 1\nunsolved goals\ncase h\nh1 : 12 ≥ 6\nh2 : Even 12\nhn : -12 =  
6\nhn8  
: -12 = 8\nhn_ge_10 : 12 ≥ 10\nhn10 : -12 = 10\nh- ∀ (q : ℕ), 3 ≤ q → q ≤ 3 → -4 % q = 0 ∧ -4 %  
q =  
12 % q","l34c3-l36c24, severity: 1\nunsolved goals\ncase h\nh1 : 14 ≥ 6\nh2 : Even 14\nhn : -14  
=  
6\nhn8 : -14 = 8\nhn_ge_10 : 14 ≥ 10\nhn10 : -14 = 10\nhn12 : -14 = 12\nh- ∀ (q : ℕ), 3 ≤ q → q  
≤ 3  
→ -4 % q = 0 ∧ -4 % q = 14 % q","l38c3-l40c24, severity: 1\nunsolved goals\ncase h\nh1 : 16 ≥  
6\nh2
```

```
: Even 16\nhn : -16 = 6\nhn8 : -16 = 8\nhn_ge_10 : 16 ≥ 10\nhn10 : -16 = 10\nhn12 : -16 = 12\nhn14 : -16 = 14\nh- ∀ (q : ℕ), 3 ≤ q → q ≤ 4 → -5 % q = 0 ∧ -5 % q = 16 % q"]}]
```

## Étape 51 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Good, now I need to finish these cases. Let me work on them one by one:

## Étape 52 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 53 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l6c9-l6c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 54 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Great! Now I need to handle  $n \geq 18$ . Let me check what values work for larger  $n$ . Actually, let me think about this more systematically. For very large  $n$ , maybe there's a general pattern. But given the complexity, let me continue with case analysis for a few more values, then see if there's a pattern.

Let me check  $n = 18, 20, 22$ , etc.:

## Étape 55 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 56 : Thinking (Réflexion)

### Pensée de l'Agent Numina

This is getting too complicated. Let me try a different approach. Maybe I should use a more general construction. Looking at the problem again, let me think about what mathematical property this is capturing.

Actually, I realize this might not be a standard theorem, and the specific values might matter. Let me try a different approach: use classical logic to assert existence and then prove the properties computationally for specific ranges.

Let me restart with a cleaner approach:

## Étape 57 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 58 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n      ¬(p % q = 0) ∧ ¬(p % q = n % q) := by\nBefore:\nNo goals at line\nstart.\nAfter:\nn : ℕ\nh1 : n ≥ 6\nh2 : Even n\n¬ ∃ p, 3 ≤ p ∧ p ≤ n / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ n.sqrt → ¬p % q = 0 ∧ ¬p % q = n % q"}
```

## Étape 59 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me actually think about this problem differently. Maybe the statement is malformed or impossible to prove as stated. Let me double-check by manually verifying for  $n = 18$ :

$n = 18$ ,  $\sqrt{18} \approx 4.24$ , so  $q \in \{3, 4\}$

- Need  $p$  such that:

-  $3 \leq p \leq 9$

-  $p \% 3 \neq 0$  and  $p \% 3 \neq 18 \% 3 = 0$ , so  $p \% 3 \in \{1, 2\}$

-  $p \% 4 \neq 0$  and  $p \% 4 \neq 18 \% 4 = 2$ , so  $p \% 4 \in \{1, 3\}$

Combined:  $p \% 3 \in \{1, 2\}$  and  $p \% 4 \in \{1, 3\}$

-  $p \% 3 = 1$ ,  $p \% 4 = 1$ :  $p \equiv 1 \pmod{12}$ , so  $p \in \{4, 7\}$  (in range  $[3,9]$ ):  $p = 4$  or  $7$

-  $p \% 3 = 1$ ,  $p \% 4 = 3$ :  $p \equiv 7 \pmod{12}$ , so  $p = 7$

-  $p \% 3 = 2$ ,  $p \% 4 = 1$ :  $p \equiv 5 \pmod{12}$ , so  $p = 5$

-  $p \% 3 = 2$ ,  $p \% 4 = 3$ :  $p \equiv 11 \pmod{12}$ , but  $11 > 9$ , so none in range

So  $p \in \{4, 5, 7\}$  work for  $n = 18$ . Let me use  $p = 5$  as a general choice.

Actually, looking at the pattern, it seems like  $p = 5$  or  $p = 4$  often work. Let me try a simpler formula:

## Étape 60 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 61 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l6c9-l6c29, severity: 2\ndeclaration uses 'sorry'"]}
```

## Étape 62 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n      · norm_num\nBefore:\ncase h.«3»\nn : ℕ\nh1 : 16 ≥ 6\nh2 : Even 16\nq : ℕ\nhq_ge : 3 ≤ 3\nhq_le : 3 ≤ 4\n! -5 % 3 = 0 ∧ -5 % 3 = 16 % 3\n---\ncase h.«4»\nn : ℕ\nh1 : 16 ≥ 6\nh2 : Even 16\nq : ℕ\nhq_ge : 3 ≤ 4\nhq_le : 4 ≤ 4\n! -5 % 4 = 0 ∧ -5 % 4 = 16 % 4\nAfter:\nno goals"}
```

## Étape 63 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n | _ =>\nBefore:\nn xt : ℕ\nh1 : xt ≥ 6\nh2 : Even xt\n! - ∃ p, 3 ≤ p ∧ p ≤ xt / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ xt.sqrt → -p % q = 0 ∧ -p % q = xt % q\nAfter:\nn xt : ℕ\nh1 : xt ≥ 6\nh2 : Even xt\n! - ∃ p, 3 ≤ p ∧ p ≤ xt / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ xt.sqrt → -p % q = 0 ∧ -p % q = xt % q"}
```

## Étape 64 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Now I need to handle the general case for  $n \geq 18$ . Let me think about this more carefully. I'll try to use  $p = 5$  for most cases and verify it works:

## Étape 65 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 66 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["\u0013-\u0015, severity: 1\nUnknown identifier `n`","153c7-153c15, severity: 1\nTactic `cases` failed: major premise type is not an inductive type\n ?m.497\n\nExplanation: the `cases` tactic is for constructor-based reasoning as well as for applying custom cases principles with a `using` clause or a registered '@[cases_eliminator]' theorem. The above type neither is an inductive type nor has a registered theorem.\n\ncase succ\nn n\u2081 : \u2116\nh1 : n\u2081 + 1 + 1 \u2265 6\nh2 : Even (n\u2081 + 1)\nx\u2081 : ?m.497\nn\u2081 + 1 + 1 \u2265 17","196c37-196c67, severity: 1\nUnknown constant `Nat.eq_one_or_eq_self_of_dvd_prime`","197c9-197c14, severity: 1\n\u03c9 could not prove the goal:\na possible counterexample may satisfy the constraints\n b \u2265 5\n a \u2265 17\nwhere\n a := \u2113\n b := \u2113(n\u2081 + 1).sqrt"]}
```

## Étape 67 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Let me fix the errors:

## Étape 68 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_loogle

### Réponse du Serveur Lean

```
{"result":"No results found."}
```

## Étape 69 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_loogle

### Réponse du Serveur Lean

```
{"result":"No results found."}
```

## Étape 70 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_local\_search

### Réponse du Serveur Lean

```
{"result":[{"name":"Prime.eq_of_factorization_pos","kind":"theorem","file":".lake[workspace]/Basic.lean"}, {"name":"Prime.eq_two_or_odd","kind":"theorem","file":".lake[workspace]/Basic.lean"}, {"name":"Prime.eq_two_or_odd'","kind":"theorem","file":".lake[workspace]/Basic.lean"}, {"name":"Prime.eq_one_of_pow","kind":"theorem","file":".lake[workspace]/Basic.lean"}, {"name":"Prime.eq_one_or_self_of_dvd","kind":"theorem","file":".lake[workspace]/Defs.lean"}]}
```

## Étape 71 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Perfect! It's `Prime.eq\_one\_or\_self\_of\_dvd`. Let me fix the proof:

## Étape 72 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.

## Étape 73 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_diagnostic\_messages

### Réponse du Serveur Lean

```
{"result":["l50c34-l50c39, severity: 1\nomega could not prove the goal:\na possible counterexample may satisfy the constraints\n b ≥ 0\n a ≥ 5\n a - b ≥ 0\nwhere\n a := ↑m\n b := ↑n", "l53c7-l53c12, severity: 1\nomega could not prove the goal:\na possible counterexample may satisfy the constraints\n 3 ≤ a ≤ 8\nwhere\n a := ↑k", "l64c9-l64c14, severity: 1\nomega could not prove the goal:\na possible counterexample may satisfy the constraints\n b ≥ 5\n a ≥ 18\nwhere\n a := ↑n\n b := ↑(m + 1).sqrt"]}]}
```

## Étape 74 : Thinking (Réflexion)

### Pensée de l'Agent Numina

I see the issues. Let me fix them:

## Étape 75 : Serveur Lean LSP - mcp\_\_lean-lsp\_\_lean\_goal

### Réponse du Serveur Lean

```
{"result":"Goals on line:\n      have hn_eq : n = m + 1 := by omega\nBefore:\nn m : ℕ\nh1 : m + 1 ≥  
6\nh2 : Even (m + 1)\n¬ ∃ p, 3 ≤ p ∧ p ≤ (m + 1) / 2 ∧ ∀ (q : ℕ), 3 ≤ q ∧ q ≤ (m + 1).sqrt → ¬p % q  
= 0 ∧ ¬p % q = (m + 1) % q\nAfter:\nno goals"}
```

## Étape 76 : Thinking (Réflexion)

### Pensée de l'Agent Numina

Actually,  $n$  is already  $m + 1$  in this context. Let me simplify:

## Étape 77 : Appel d'Outil - Edit

### Interaction Système : Edit

The file [workspace]/[session].lean has been updated successfully.