

Experiment ‘p5_3_firstDigit’ Results

December 18, 2025

Experiment outcome: SUCCESS

Bad responses: 0

Responses containing assume : 0

Resolution attempts: 1

Hard fails (resolution): 0

Soft fails (resolution): 0

Verification attempts: 1

Problem Specification

Problem name: p5_3_firstDigit

Natural language statement: Write a method returning the first digit of the argument.

Method signature: p5_3_firstDigit(n: int) returns (first: int)

Ensures

- $0 \leq \text{first} < 10$
- $\text{intToSeq}(n)[0] == \text{first}$

Requires

- $0 \leq n$

Functional Code Given

```
function intToSeq(n: int) : seq<int>
    requires n >= 0
    ensures forall d :: d in intToSeq(n) ==> 0 <= d < 10
{
    if n < 10 then [n]
    else intToSeq(n / 10) + [n % 10]
}
```

GenAI interactions

Below you will find all interactions between the ‘user’ (program) and the ‘assistant’ (OpenAI).

Program → GenAI

You are given the following task to perform in Dafny:

Write a method returning the first digit of the argument.

The signature should be:

```
method p5_3_firstDigit(n: int) returns (first: int)
```

The method should respect the following contract:

```
requires 0 <= n, ensures 0 <= first < 10, ensures intToSeq(n)[0] == first
```

The contract uses the following dafny code:

```
function intToSeq(n: int) : seq<int>
  requires n >= 0
  ensures forall d :: d in intToSeq(n) ==> 0 <= d < 10
{
  if n < 10 then [n]
  else intToSeq(n / 10) + [n % 10]
}
```

These function(s) must not be used in your implementation of the method.

Produce and show only the Dafny body of this method, including the curly braces that
 → surround it. Do not show the signature nor contract. You must not use ‘assume’
 → anywhere in your code.

GenAI → Program

System fingerprint:

ID: chatcmpl-Co4tg8JuTVT15L7Pp977x76N6VWKY

Model: gpt-5.1-chat-latest

Created at: 1766051260

Finish reason: stop

Usage: 462 tokens in, and 313 tokens out

```
{
  if n < 10 {
    first := n;
  } else {
    first := p5_3_firstDigit(n / 10);
  }
}
```

Final Program

```
function intToSeq(n: int) : seq<int>
    requires n >= 0
    ensures forall d :: d in intToSeq(n) ==> 0 <= d < 10
{
    if n < 10 then [n]
    else intToSeq(n / 10) + [n % 10]
}

method p5_3_firstDigit(n: int) returns (first: int)
    requires 0 <= n
    ensures 0 <= first < 10
    ensures intToSeq(n)[0] == first
{
    if n < 10 {
        first := n;
    } else {
        first := p5_3_firstDigit(n / 10);
    }
}
```

Total Token Usage

Input tokens: 462

Output tokens: 313

Reasoning tokens: 256

Sum of ‘total tokens’: 775

Experiment Timings

Overall Experiment started at 1766051261986, ended at 1766051267884, lasting 5898ms (5.90 seconds)

Iteration #1 started at 1766051261987, ended at 1766051267884, lasting 5897ms (5.90 seconds)