

Experiment ‘p5_4_middle’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY

Bad responses: 0

Responses containing assume : 0

Resolution attempts: 8

Hard fails (resolution): 0

Soft fails (resolution): 0

Verification attempts: 8

Problem Specification

Problem name: p5_4_middle

Natural language statement: Write a method that returns the middle character of a string if the length is odd, or the two middle characters if the length is even. For example, middle("middle") returns "dd".

Method signature: p5_4_middle(str: seq<char>) returns (middleStr: seq<char>)

Ensures

- if $|str| \% 2 == 1$ then $|middleStr| == 1$ else $|middleStr| == 2$
- if $|str| \% 2 == 1$ then $middleStr[0] == str[|str| / 2]$ else $middleStr == str[|str| / 2 - 1 .. |str| / 2]$

Requires

- $0 < |str|$

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 that returns the middle character of a string if the length is odd, or the
→ two middle characters if the length is even. For example, `middle("middle")`
→ returns "dd".

The signature should be:

```
method p5_4_middle(str: seq<char>) returns (middleStr: seq<char>)
```

The method should respect the following contract:

```
requires 0 < |str|, ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| ==  
→ 2, ensures if |str| % 2 == 1 then middleStr[0] == str[|str| / 2] else middleStr ==  
→ str[|str| / 2 - 1 .. |str| / 2]
```

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: chatmpl-Co4tnyp3EoRS2mrYikDMTbOy66Efb

Model: gpt-5.1-chat-latest

Created at: 1766051267

Finish reason: stop

Usage: 458 tokens in, and 142 tokens out

```
{  
    var n := |str|;  
    if n % 2 == 1 {  
        middleStr := [str[n/2]];  
    } else {  
        middleStr := str[n/2 - 1 .. n/2];  
    }  
}
```

Program → GenAI

When using `dafny verify`, the below error is emitted and verify fails:

```
problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on  
→ this return path  
|
```

```

5 | {
| ^
problem_data/p5_4_middle/program.dfy(3,66): Related location: this is the postcondition
  ↪ that could not be proved
|
3 | ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| == 2
| ^^

Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
  ↪ 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-Co4tqqXy2aL99hCMPbv1044Yh2YBg
Model: gpt-5.1-chat-latest
Created at: 1766051270
Finish reason: stop
Usage: 696 tokens in, and 145 tokens out

```
{
  var n := |str|;
  if n % 2 == 1 {
    middleStr := [str[n/2]];
  } else {
    middleStr := str[n/2 - 1 .. n/2 + 1];
  }
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```

problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on
  ↪ this return path
|
5 | {
| ^
problem_data/p5_4_middle/program.dfy(4,78): Related location: this is the postcondition
  ↪ that could not be proved
|
4 | ensures if |str| % 2 == 1 then middleStr[0] == str[|str| / 2] else middleStr == str
  ↪ [|str| / 2 - 1 .. |str| / 2]
| ^^

```

```
Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
↳ 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: chatmpl-Co4ttCTIgN35HFysKg4qZZjGiTIwy

Model: gpt-5.1-chat-latest

Created at: 1766051273

Finish reason: stop

Usage: 1007 tokens in, and 206 tokens out

```
{
    var n := |str|;
    if n % 2 == 1 {
        middleStr := [str[n/2]];
    } else {
        middleStr := str[n/2 - 1 .. n/2];
    }
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```
problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on
↳ this return path
|
5 | {
| ^
|
problem_data/p5_4_middle/program.dfy(3,66): Related location: this is the postcondition
↳ that could not be proved
|
3 | ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| == 2
| ^~
```

```
Dafny program verifier finished with 1 verified, 1 error
```

```
Correct the error by altering only the method body. Produce and show only the Dafny body,
↳ 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-Co4tzlI4OXkISsyB5pRtRUz82eqA1

Model: gpt-5.1-chat-latest

Created at: 1766051279

Finish reason: stop

Usage: 1249 tokens in, and 334 tokens out

```
{  
    var n := |str|;  
    if n % 2 == 1 {  
        middleStr := [str[n/2]];  
    } else {  
        middleStr := str[n/2 - 1 .. n/2];  
    }  
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```
problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on  
    ↪ this return path  
|  
5 | {  
| ^  
  
problem_data/p5_4_middle/program.dfy(3,66): Related location: this is the postcondition  
    ↪ that could not be proved  
|  
3 | ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| == 2  
| ^^
```

Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
 ↪ 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-Co4u5NIsIhwTx6aRFXd1NOaQWZRte

Model: gpt-5.1-chat-latest

Created at: 1766051285

Finish reason: stop

Usage: 1513 tokens in, and 400 tokens out

```
{  
    var n := |str|;
```

```

if n % 2 == 1 {
    middleStr := [str[n/2]];
} else {
    middleStr := [str[n/2 - 1], str[n/2]];
}
}

```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```

problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on
    ↪ this return path
|
5 | {
| ^
|
problem_data/p5_4_middle/program.dfy(4,78): Related location: this is the postcondition
    ↪ that could not be proved
|
4 | ensures if |str| % 2 == 1 then middleStr[0] == str[|str| / 2] else middleStr == str
    ↪ [|str| / 2 - 1 .. |str| / 2]
| ^^

```

Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
 ↪ 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: chatmpl-Co4uCc4CFVd19mnxAaGXdEMzo0fN

Model: gpt-5.1-chat-latest

Created at: 1766051292

Finish reason: stop

Usage: 1823 tokens in, and 142 tokens out

```
{
    var n := |str|;
    if n % 2 == 1 {
        middleStr := [str[n/2]];
    } else {
        middleStr := str[n/2 - 1 .. n/2];
    }
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```
problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on
  ↪ this return path
|
5 | {
| ^
|
problem_data/p5_4_middle/program.dfy(3,66): Related location: this is the postcondition
  ↪ that could not be proved
|
3 | ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| == 2
| ^~
```

Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
↪ 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: chatmpl-Co4uGcQGkM4JeIYcCuvE9qknZte56

Model: gpt-5.1-chat-latest

Created at: 1766051296

Finish reason: stop

Usage: 2065 tokens in, and 409 tokens out

```
{
    var n := |str|;
    if n % 2 == 1 {
        middleStr := [str[n/2]];
    } else {
        // Must match the postcondition's exact slice
        middleStr := str[n/2 - 1 .. n/2];
    }
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```
problem_data/p5_4_middle/program.dfy(5,0): Error: a postcondition could not be proved on
  ↪ this return path
|
5 | {
| ^~
```

```

problem_data/p5_4_middle/program.dfy(3,66): Related location: this is the postcondition
  ↳ that could not be proved
  |
3 | ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| == 2
| ^

Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
  ↳ 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: chatmpl-Co4uMpQzfReXHtKjLZBaYkfdPUNOV

Model: gpt-5.1-chat-latest

Created at: 1766051302

Finish reason: stop

Usage: 2340 tokens in, and 289 tokens out

```
{
  var n := |str|;
  if n % 2 == 1 {
    middleStr := [str[n/2]];
  } else {
    // To satisfy both postconditions, the slice must have length 2
    middleStr := str[n/2 - 1 .. n/2 + 1];
  }
}
```

Final Program

```

method p5_4_middle(str: seq<char>) returns (middleStr: seq<char>)
  requires 0 < |str|
  ensures if |str| % 2 == 1 then |middleStr| == 1 else |middleStr| == 2
  ensures if |str| % 2 == 1 then middleStr[0] == str[|str| / 2] else middleStr ==
    ↳ str[|str| / 2 - 1 .. |str| / 2]
{
  var n := |str|;
  if n % 2 == 1 {
    middleStr := [str[n/2]];
  } else {
    // To satisfy both postconditions, the slice must have length 2
    middleStr := str[n/2 - 1 .. n/2 + 1];
  }
}

```

Total Token Usage

Input tokens: 11151

Output tokens: 2067

Reasoning tokens: 1408

Sum of ‘total tokens’: 13218

Experiment Timings

Iteration #8 started at 1766051304705, ended at 1766051310426, lasting 5721ms (5.72 seconds)

Overall Experiment started at 1766051268744, ended at 1766051310426, lasting 41682ms (41.68 seconds)

Iteration #4 started at 1766051280587, ended at 1766051287288, lasting 6701ms (6.70 seconds)

Iteration #5 started at 1766051287288, ended at 1766051294307, lasting 7019ms (7.02 seconds)

Iteration #6 started at 1766051294307, ended at 1766051297944, lasting 3637ms (3.64 seconds)

Iteration #7 started at 1766051297944, ended at 1766051304705, lasting 6761ms (6.76 seconds)

Iteration #1 started at 1766051268744, ended at 1766051272473, lasting 3729ms (3.73 seconds)

Iteration #2 started at 1766051272473, ended at 1766051275768, lasting 3295ms (3.30 seconds)

Iteration #3 started at 1766051275768, ended at 1766051280587, lasting 4819ms (4.82 seconds)