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: chatcmpl-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: chatcmpl-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: chatcmpl-Co4uCc4CFVd19mnxAAaGXdEMzo0fN
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: chatcmpl-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: chatcmpl-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)

Experiment ‘p5_5_repeat’ 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): 1
Verification attempts: 7

Problem Specification

Problem name: p5_5_repeat
Natural language statement: Write a method that returns a string repeated n times. For example, repeat(”ho”, 3) returns ”hohoho”.
Method signature: p5_5_repeat(str: seq<char>, n: int) returns (repeatedStr: seq<char>)

Ensures

- `|repeatedStr| == n * |str|`
- `forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] == str`

Requires

- `n >= 0`

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 a string repeated n times. For example, repeat("ho", 3)
    ↪ returns "hohoho".

The signature should be:

method p5_5_repeat(str: seq<char>, n: int) returns (repeatedStr: seq<char>)

The method should respect the following contract:

requires n >= 0, ensures |repeatedStr| == n * |str|, ensures forall i :: 0 <= i < n ==>
    ↪ repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] == str

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-Co4uTxy9ICWgiKtzGLgBGUx4bvIk3
Model: gpt-5.1-chat-latest
Created at: 1766051309
Finish reason: stop
Usage: 382 tokens in, and 262 tokens out

```
{
  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Program → GenAI

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

```
problem_data/p5_5_repeat/program.dfy(4,47): Error: upper bound below lower bound or above
    ↪  length of sequence
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^

problem_data/p5_5_repeat/program.dfy(11,14): Error: this invariant could not be proved to
    ↪  be maintained by the loop
 Related message: loop invariant violation
   |
11 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
   | ^^^^^^

problem_data/p5_5_repeat/program.dfy(11,44): Error: upper bound below lower bound or
    ↪ above length of sequence
   |
11 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
   | ^


Dafny program verifier finished with 0 verified, 3 errors

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-Co4uYRoxqFZxZTbl6cssZ5CMUGdv1
Model: gpt-5.1-chat-latest
Created at: 1766051314
Finish reason: stop
Usage: 964 tokens in, and 131 tokens out

```
{
  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str|] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Program → GenAI

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

problem_data/p5_5_repeat/program.dfy(4,47): Error: upper bound below lower bound or above
    ↪  length of sequence
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^

problem_data/p5_5_repeat/program.dfy(5,0): Error: a postcondition could not be proved on
    ↪ this return path
  |
5 | {
  | ^

problem_data/p5_5_repeat/program.dfy(4,9): Related location: this is the postcondition
    ↪ that could not be proved
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^^^^^^

problem_data/p5_5_repeat/program.dfy(11,14): Error: this invariant could not be proved to
    ↪  be maintained by the loop
 Related message: loop invariant violation
   |
11 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str|] == str
   | ^^^^^^


Dafny program verifier finished with 0 verified, 3 errors

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-Co4v3Y7jSItgi7bvoG0ZojRZtUCei
Model: gpt-5.1-chat-latest
Created at: 1766051345
Finish reason: stop
Usage: 1488 tokens in, and 396 tokens out

```
{
  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |str| >= 0
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str|] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Program → GenAI

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

problem_data/p5_5_repeat/program.dfy(4,47): Error: upper bound below lower bound or above
    ↪  length of sequence
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^

problem_data/p5_5_repeat/program.dfy(5,0): Error: a postcondition could not be proved on
    ↪ this return path
  |
5 | {
  | ^

problem_data/p5_5_repeat/program.dfy(4,9): Related location: this is the postcondition
    ↪ that could not be proved
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^^^^^^


Dafny program verifier finished with 0 verified, 2 errors

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-Co4vd8a6idnToh4i1C0YAnHlyb1dJ
Model: gpt-5.1-chat-latest
Created at: 1766051381
Finish reason: stop
Usage: 1820 tokens in, and 134 tokens out

```
{
  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Program → GenAI

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

problem_data/p5_5_repeat/program.dfy(4,47): Error: upper bound below lower bound or above
    ↪  length of sequence
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^

problem_data/p5_5_repeat/program.dfy(11,14): Error: this invariant could not be proved to
    ↪  be maintained by the loop
 Related message: loop invariant violation
   |
11 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
   | ^^^^^^

problem_data/p5_5_repeat/program.dfy(11,44): Error: upper bound below lower bound or
    ↪ above length of sequence
   |
11 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
   | ^


Dafny program verifier finished with 0 verified, 3 errors

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-Co4vhKeV31MJkaCcRNCtI2LBMv0sc
Model: gpt-5.1-chat-latest
Created at: 1766051385

Finish reason: stop
Usage: 2304 tokens in, and 820 tokens out

```
{
  // Handle the special case |str| == 0 separately: the postcondition becomes vacuous
  if |str| == 0 {
    repeatedStr := [];
    return;
  }

  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Program → GenAI

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

problem_data/p5_5_repeat/program.dfy(4,47): Error: upper bound below lower bound or above
    ↪  length of sequence
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^

problem_data/p5_5_repeat/program.dfy(9,4): Error: a postcondition could not be proved on
    ↪ this return path
  |
9 | return;
  | ^^^^^^^

problem_data/p5_5_repeat/program.dfy(4,9): Related location: this is the postcondition
    ↪ that could not be proved
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^^^^^^

problem_data/p5_5_repeat/program.dfy(17,14): Error: this invariant could not be proved to
    ↪  be maintained by the loop
 Related message: loop invariant violation
   |
17 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
   | ^^^^^^


Dafny program verifier finished with 0 verified, 3 errors

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-Co4vsTLqoWNc6cGGBKOBGXOe78FSD
Model: gpt-5.1-chat-latest
Created at: 1766051396
Finish reason: stop
Usage: 2891 tokens in, and 751 tokens out

```
{
  var res: seq<char> := [];
  var i: int := 0;

  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
    invariant forall j :: 0 <= j < i ==> j * |str| <= (j + 1) * |str| - 1 + 1 <= |res|
  {
    res := res + str;
    i := i + 1;
  }

  repeatedStr := res;
}
```

Program → GenAI

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

problem_data/p5_5_repeat/program.dfy(13,14): Warning: Could not find a trigger for this
    ↪ quantifier. Without a trigger, the quantifier may cause brittle verification. To
    ↪ silence this warning, add an explicit trigger using the {:trigger} attribute. For
    ↪ more information, see the section on quantifier instantiation rules in the
    ↪ reference manual.
   |
13 | invariant forall j :: 0 <= j < i ==> j * |str| <= (j + 1) * |str| - 1 + 1 <= |res|
   | ^^^^^^

Compilation failed because warnings were found and --allow-warnings is false
```

```
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-Co4w2jp6NntgBZwvnExsP5JAsLh1h
Model: gpt-5.1-chat-latest
Created at: 1766051406
Finish reason: stop
Usage: 3105 tokens in, and 264 tokens out

```
{
  var res: seq<char> := [];
  var i: int := 0;

  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
  {
    res := res + str;
    i := i + 1;
  }

  repeatedStr := res;
}
```

Program → GenAI

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

problem_data/p5_5_repeat/program.dfy(4,47): Error: upper bound below lower bound or above
    ↪  length of sequence
  |
4 |  ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ] ==
    ↪  str
  | ^

problem_data/p5_5_repeat/program.dfy(12,14): Error: this invariant could not be proved to
    ↪  be maintained by the loop
 Related message: loop invariant violation
   |
12 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
   | ^^^^^^

problem_data/p5_5_repeat/program.dfy(12,44): Error: upper bound below lower bound or
    ↪ above length of sequence
   |
12 | invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
```

```
  | ^


Dafny program verifier finished with 0 verified, 3 errors

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-Co4w64T4rFfWNHqFN4GJtsJTziMJ5
Model: gpt-5.1-chat-latest
Created at: 1766051410
Finish reason: stop
Usage: 3664 tokens in, and 713 tokens out

```
{
  // Only valid way to satisfy the given postcondition when |str| can be 0
  // is to ensure that the verifier knows n = 0 in that case.
  if |str| == 0 {
    // If |str| == 0 but n > 0, the postcondition's slices are invalid.
    // Therefore we construct repeatedStr normally but also prove n == 0.
    // Since |str| == 0, n * |str| == 0, so repeatedStr must be empty.
    repeatedStr := [];
    return;
  }

  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Final Program

```
method p5_5_repeat(str: seq<char>, n: int) returns (repeatedStr: seq<char>)
        requires n >= 0
        ensures |repeatedStr| == n * |str|
        ensures forall i :: 0 <= i < n ==> repeatedStr[i * |str| .. (i + 1) * |str| - 1 ]
            ↪ == str
{
  // Only valid way to satisfy the given postcondition when |str| can be 0
  // is to ensure that the verifier knows n = 0 in that case.
  if |str| == 0 {
    // If |str| == 0 but n > 0, the postcondition's slices are invalid.
    // Therefore we construct repeatedStr normally but also prove n == 0.
    // Since |str| == 0, n * |str| == 0, so repeatedStr must be empty.
    repeatedStr := [];
    return;
  }

  var res: seq<char> := [];
  var i: int := 0;
  while i < n
    invariant 0 <= i <= n
    invariant |res| == i * |str|
    invariant forall j :: 0 <= j < i ==> res[j * |str| .. (j + 1) * |str| - 1] == str
  {
    res := res + str;
    i := i + 1;
  }
  repeatedStr := res;
}
```

Total Token Usage

Input tokens: 16618
Output tokens: 3471
Reasoning tokens: 2176
Sum of 'total tokens': 20089

Experiment Timings

Iteration #8 started at 1766051412214, ended at 1766051422291, lasting 10077ms (10.08 seconds)
Overall Experiment started at 1766051311146, ended at 1766051422291, lasting 111145ms (111.15 seconds)
Iteration #4 started at 1766051383316, ended at 1766051387131, lasting 3815ms (3.82 seconds)
Iteration #5 started at 1766051387131, ended at 1766051398368, lasting 11237ms (11.24 seconds)
Iteration #6 started at 1766051398368, ended at 1766051407892, lasting 9524ms (9.52 seconds)
Iteration #7 started at 1766051407892, ended at 1766051412214, lasting 4322ms (4.32 seconds)
Iteration #1 started at 1766051311146, ended at 1766051315935, lasting 4789ms (4.79 seconds)
Iteration #2 started at 1766051315935, ended at 1766051347226, lasting 31291ms (31.29 seconds)
Iteration #3 started at 1766051347227, ended at 1766051383316, lasting 36089ms (36.09 seconds)

Experiment ‘p5_6_countVowels’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY
Bad responses: 0
Responses containing `assume` **:** 0
Resolution attempts: 8
Hard fails (resolution): 6
Soft fails (resolution): 0
Verification attempts: 2

Problem Specification

Problem name: p5_6_countVowels
Natural language statement: Write a method that returns the count of all vowels in a string. Vowels are the letters a, e, i, o, and u, and their uppercase variants.
Method signature: p5_6_countVowels(str: seq<char>) returns (vowelCount: int)

Ensures

- `vowelCount >= 0`
- `vowelCount <= |str|`
- `(forall c :: c in str ==> (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==> vowelCount == |str|`
- `(forall c :: c in str ==> !(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==> vowelCount == 0`

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 count of all vowels in a string. Vowels are the letters a
    ↪ , e, i, o, and u, and their uppercase variants.

The signature should be:

method p5_6_countVowels(str: seq<char>) returns (vowelCount: int)

The method should respect the following contract:

ensures vowelCount >= 0, ensures vowelCount <= |str|, ensures (forall c :: c in str ==> (
    ↪ c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' || c == 'A' || c == 'E'
    ↪ || c == 'I' || c == 'O' || c == 'U')) ==> vowelCount == |str|, ensures (forall c
    ↪ :: c in str ==> !(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' || c ==
    ↪  'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==> vowelCount == 0

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-Co4wH07wVDD77id34cPignYubvwUL
Model: gpt-5.1-chat-latest
Created at: 1766051421
Finish reason: stop
Usage: 618 tokens in, and 477 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|;
    invariant 0 <= vowelCount <= i;
    invariant vowelCount == |{ j:int | 0 <= j < i && (str[j] == 'a' || str[j] == 'e' ||
        ↪ str[j] == 'i' || str[j] == 'o' || str[j] == 'u' || str[j] == 'A' || str[j] == '
        ↪ E' || str[j] == 'I' || str[j] == 'O' || str[j] == 'U') }|;
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
       str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(10,29): Warning: deprecated style: a semi-colon
    ↪  is not needed here
   |
10 | invariant 0 <= i <= |str|;
   | ^

problem_data/p5_6_countVowels/program.dfy(11,34): Warning: deprecated style: a semi-colon
    ↪  is not needed here
   |
11 | invariant 0 <= vowelCount <= i;
   | ^

problem_data/p5_6_countVowels/program.dfy(12,32): Error: rbrace expected
   |
12 | invariant vowelCount == |{ j:int | 0 <= j < i && (str[j] == 'a' || str[j] == 'e' ||
    ↪ str[j] == 'i' || str[j] == 'o' || str[j] == 'u' || str[j] == 'A' || str[j] == 'E'
    ↪ || str[j] == 'I' || str[j] == 'O' || str[j] == 'U') }|;
   | ^

1 parse errors detected in program.dfy

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-Co4wNyixqZWUlCplf3uBni0mSosHh
Model: gpt-5.1-chat-latest
Created at: 1766051427
Finish reason: stop
Usage: 1219 tokens in, and 544 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
    invariant vowelCount == |({ j:int | 0 <= j < i &&
                                (str[j] == 'a' || str[j] == 'e' || str[j] == 'i' || str[j] ==
                                    ↪  'o' || str[j] == 'u' ||
                               str[j] == 'A' || str[j] == 'E' || str[j] == 'I' || str[j] ==
                                   ↪  'O' || str[j] == 'U') })|
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
       str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(12,33): Error: rbrace expected
   |
12 | invariant vowelCount == |({ j:int | 0 <= j < i &&
   | ^

1 parse errors detected in program.dfy

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-Co4wUPCwOvYEEPvCKOOdol8LjcOi4
Model: gpt-5.1-chat-latest
Created at: 1766051434
Finish reason: stop
Usage: 1447 tokens in, and 352 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
    invariant vowelCount == |{ j:int | 0 <= j < i &&
                            (str[j] == 'a' || str[j] == 'e' || str[j] == 'i' || str[j] ==
                                ↪ 'o' || str[j] == 'u' ||
                             str[j] == 'A' || str[j] == 'E' || str[j] == 'I' || str[j] == '
                                 ↪ O' || str[j] == 'U') }|
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
      str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(12,32): Error: rbrace expected
   |
12 | invariant vowelCount == |{ j:int | 0 <= j < i &&
   | ^

1 parse errors detected in program.dfy

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-Co4wY5mpGoQpmdQX67nEJBXm2v3lV
Model: gpt-5.1-chat-latest
Created at: 1766051438
Finish reason: stop
Usage: 1857 tokens in, and 246 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
      str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(6,0): Error: a postcondition could not be
    ↪ proved on this return path
  |
6 | {
  | ^

problem_data/p5_6_countVowels/program.dfy(4,170): Related location: this is the
    ↪ postcondition that could not be proved
  |
4 |  ensures (forall c :: c in str ==> (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c
    ↪  == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==>
    ↪ vowelCount == |str|
  | ^^

problem_data/p5_6_countVowels/program.dfy(6,0): Error: a postcondition could not be
    ↪ proved on this return path
  |
6 | {
  | ^

problem_data/p5_6_countVowels/program.dfy(5,171): Related location: this is the
    ↪ postcondition that could not be proved
  |
5 |  ensures (forall c :: c in str ==> !(c == 'a' || c == 'e' || c == 'i' || c == 'o' ||
    ↪ c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==>
    ↪ vowelCount == 0
  | ^^


Dafny program verifier finished with 0 verified, 2 errors

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-Co4wdZpf278cUlDEPJGwsWHKNiiEa
Model: gpt-5.1-chat-latest
Created at: 1766051443
Finish reason: stop
Usage: 2713 tokens in, and 282 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
    invariant vowelCount == |{ j:int | 0 <= j < i && (str[j] == 'a' || str[j] == 'e' ||
         ↪ str[j] == 'i' || str[j] == 'o' || str[j] == 'u' || str[j] == 'A' || str[j] == '
         ↪ E' || str[j] == 'I' || str[j] == 'O' || str[j] == 'U') }|
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
         ↪ ||
       str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(12,32): Error: rbrace expected
   |
12 | invariant vowelCount == |{ j:int | 0 <= j < i && (str[j] == 'a' || str[j] == 'e' ||
    ↪ str[j] == 'i' || str[j] == 'o' || str[j] == 'u' || str[j] == 'A' || str[j] == 'E'
    ↪ || str[j] == 'I' || str[j] == 'O' || str[j] == 'U') }|
   | ^

1 parse errors detected in program.dfy

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-Co4wiCz2Cw9KGbDTpSC6cmeURaWSu
Model: gpt-5.1-chat-latest
Created at: 1766051448
Finish reason: stop
Usage: 3005 tokens in, and 546 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
    invariant vowelCount == (| { j:int | 0 <= j < i &&
                             (str[j] == 'a' || str[j] == 'e' || str[j] == 'i' || str[j] ==
                                 ↪ 'o' || str[j] == 'u' ||
                              str[j] == 'A' || str[j] == 'E' || str[j] == 'I' || str[j] ==
                                  ↪  'O' || str[j] == 'U') } |)
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
       str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(12,34): Error: rbrace expected
   |
12 | invariant vowelCount == (| { j:int | 0 <= j < i &&
   | ^

1 parse errors detected in program.dfy

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-Co4wpheKtNSDCyi1rro0pCLMNIYZJ
Model: gpt-5.1-chat-latest
Created at: 1766051455
Finish reason: stop
Usage: 3337 tokens in, and 182 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
       str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_6_countVowels/program.dfy(6,0): Error: a postcondition could not be
    ↪ proved on this return path
  |
6 | {
  | ^

problem_data/p5_6_countVowels/program.dfy(4,170): Related location: this is the
    ↪ postcondition that could not be proved
  |
4 |  ensures (forall c :: c in str ==> (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c
    ↪  == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==>
    ↪ vowelCount == |str|
  | ^^

problem_data/p5_6_countVowels/program.dfy(6,0): Error: a postcondition could not be
    ↪ proved on this return path
  |
6 | {
  | ^

problem_data/p5_6_countVowels/program.dfy(5,171): Related location: this is the
    ↪ postcondition that could not be proved
  |
5 |  ensures (forall c :: c in str ==> !(c == 'a' || c == 'e' || c == 'i' || c == 'o' ||
    ↪ c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==>
    ↪ vowelCount == 0
  | ^^


Dafny program verifier finished with 0 verified, 2 errors

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-Co4wtqGduhREERH6MB0zfz6TisqzD
Model: gpt-5.1-chat-latest
Created at: 1766051459
Finish reason: stop
Usage: 4192 tokens in, and 540 tokens out

```
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
    invariant vowelCount == (|{ j:int | 0 <= j < i && (str[j] == 'a' || str[j] == 'e' ||
        ↪ str[j] == 'i' || str[j] == 'o' || str[j] == 'u' || str[j] == 'A' || str[j] == '
        ↪ E' || str[j] == 'I' || str[j] == 'O' || str[j] == 'U') }|)
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
      str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Final Program

```
method p5_6_countVowels(str: seq<char>) returns (vowelCount: int)
       ensures vowelCount >= 0
       ensures vowelCount <= |str|
       ensures (forall c :: c in str ==> (c == 'a' || c == 'e' || c == 'i' || c == 'o' ||
           ↪   c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')) ==>
           ↪   vowelCount == |str|
       ensures (forall c :: c in str ==> !(c == 'a' || c == 'e' || c == 'i' || c == 'o'
           ↪  || c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U'))
           ↪  ==> vowelCount == 0
{
  vowelCount := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= vowelCount <= i
    invariant vowelCount == (|{ j:int | 0 <= j < i && (str[j] == 'a' || str[j] == 'e' ||
        ↪ str[j] == 'i' || str[j] == 'o' || str[j] == 'u' || str[j] == 'A' || str[j] == '
        ↪ E' || str[j] == 'I' || str[j] == 'O' || str[j] == 'U') }|)
  {
    if str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u'
        ↪ ||
      str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U'
    {
      vowelCount := vowelCount + 1;
    }
    i := i + 1;
  }
}
```

Total Token Usage

Input tokens: 18388
Output tokens: 3169
Reasoning tokens: 1088
Sum of 'total tokens': 21557

Experiment Timings

Iteration #8 started at 1766051461212, ended at 1766051467731, lasting 6519ms (6.52 seconds)
Overall Experiment started at 1766051423027, ended at 1766051467731, lasting 44704ms (44.70 seconds)
Iteration #4 started at 1766051440767, ended at 1766051445689, lasting 4922ms (4.92 seconds)
Iteration #5 started at 1766051445689, ended at 1766051450561, lasting 4872ms (4.87 seconds)
Iteration #6 started at 1766051450561, ended at 1766051457094, lasting 6533ms (6.53 seconds)
Iteration #7 started at 1766051457094, ended at 1766051461212, lasting 4118ms (4.12 seconds)
Iteration #1 started at 1766051423027, ended at 1766051429320, lasting 6293ms (6.29 seconds)
Iteration #2 started at 1766051429320, ended at 1766051436027, lasting 6707ms (6.71 seconds)
Iteration #3 started at 1766051436027, ended at 1766051440767, lasting 4740ms (4.74 seconds)

Experiment ‘p5_7_countWords’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY
Bad responses: 0
Responses containing `assume` **:** 0
Resolution attempts: 8
Hard fails (resolution): 2
Soft fails (resolution): 0
Verification attempts: 6

Problem Specification

Problem name: p5_7_countWords
Natural language statement: Write a method that returns the count of all words in a string. Words are separated by spaces. For example, countWords(”Mary had a little lamb”) should return 5.
Method signature: p5_7_countWords(str: seq<char>) returns (wordCount: int)

Ensures

- `wordCount >= 0`
- `if |str| == 0 then wordCount == 0 else wordCount >= 1`
- `wordCount == count_space(str) + 1`

Functional Code Given

```
function count_space(str: seq<char>) : int
{
  if |str| == 0 then 0
  else if str[0] == ' ' then 1 + count_space(str[1..])
  else count_space(str[1..])
}
```

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 count of all words in a string. Words are separated by
    ↪ spaces. For example, countWords("Mary had a little lamb") should return 5.

The signature should be:

method p5_7_countWords(str: seq<char>) returns (wordCount: int)

The method should respect the following contract:

ensures wordCount >= 0, ensures if |str| == 0 then wordCount == 0 else wordCount >= 1,
    ↪ ensures wordCount == count_space(str) + 1

The contract uses the following dafny code:

function count_space(str: seq<char>) : int
{
  if |str| == 0 then 0
  else if str[0] == ' ' then 1 + count_space(str[1..])
  else count_space(str[1..])
}

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-Co4x0Tb24wdz37KG6BopJX1sHz6AH
Model: gpt-5.1-chat-latest
Created at: 1766051466
Finish reason: stop
Usage: 520 tokens in, and 343 tokens out

```
{
  if |str| == 0 {
    wordCount := 0;
  } else {
    var i := 0;
    wordCount := 1;
    while i < |str|
      invariant 0 <= i <= |str|
      invariant wordCount == count_space(str[..i]) + 1
      invariant wordCount >= 1
      decreases |str| - i
    {
      if str[i] == ' ' {
        wordCount := wordCount + 1;
      }
      i := i + 1;
    }
  }
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(12,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
12 | {
   | ^

problem_data/p5_7_countWords/program.dfy(11,19): Related location: this is the
    ↪ postcondition that could not be proved
   |
11 |  ensures wordCount == count_space(str) + 1
   | ^^

problem_data/p5_7_countWords/program.dfy(20,26): Error: this invariant could not be
    ↪ proved to be maintained by the loop
 Related message: loop invariant violation
   |
20 | invariant wordCount == count_space(str[..i]) + 1
   | ^^


Dafny program verifier finished with 1 verified, 2 errors

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-Co4x6gMze5AIP76VJF4paTZxkuvKO
Model: gpt-5.1-chat-latest
Created at: 1766051472
Finish reason: stop
Usage: 894 tokens in, and 209 tokens out

```
{
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    decreases |str| - i
  {
    if str[i] == ' ' {
      s := s + 1;
    }
    i := i + 1;
  }
  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := s + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(12,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
12 | {
   | ^

problem_data/p5_7_countWords/program.dfy(11,19): Related location: this is the
    ↪ postcondition that could not be proved
   |
11 |  ensures wordCount == count_space(str) + 1
   | ^^


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-Co4xBQd6uD9rrCHrB6xs0RjPj6JpI
Model: gpt-5.1-chat-latest
Created at: 1766051477

Finish reason: stop
Usage: 1150 tokens in, and 423 tokens out

```
{
  // Compute count_space(str) without calling count_space
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    invariant s == count_space(str[..i])
    decreases |str| - i
  {
    if str[i] == ' ' {
      s := s + 1;
    }
    i := i + 1;
  }

  // To satisfy the (inconsistent) postconditions, we must return count_space(str) + 1
  wordCount := s + 1;
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(12,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
12 | {
   | ^

problem_data/p5_7_countWords/program.dfy(10,38): Related location: this is the
    ↪ postcondition that could not be proved
   |
10 |  ensures if |str| == 0 then wordCount == 0 else wordCount >= 1
   | ^^

problem_data/p5_7_countWords/program.dfy(12,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
12 | {
   | ^

problem_data/p5_7_countWords/program.dfy(11,19): Related location: this is the
    ↪ postcondition that could not be proved
   |
11 |  ensures wordCount == count_space(str) + 1
   | ^^
```

```
problem_data/p5_7_countWords/program.dfy(19,16): Error: this invariant could not be
    ↪ proved to be maintained by the loop
 Related message: loop invariant violation
   |
19 | invariant s == count_space(str[..i])
   | ^^


Dafny program verifier finished with 1 verified, 3 errors

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-Co4xISh1ILm3ivQpb6iVovxb0KJND
Model: gpt-5.1-chat-latest
Created at: 1766051484
Finish reason: stop
Usage: 1803 tokens in, and 495 tokens out

```
{
  // First loop: compute s = number of spaces, no use of count_space in invariants
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    decreases |str| - i
  {
    if str[i] == ' ' {
      s := s + 1;
    }
    i := i + 1;
  }

  // Second loop (ghost): prove s == count_space(str)
  ghost var gs := 0;
  ghost var j := 0;
  while j < |str|
    invariant 0 <= j <= |str|
    invariant gs == count_space(str[..j])
    decreases |str| - j
  {
    if str[j] == ' ' {
      gs := gs + 1;
    }
    j := j + 1;
  }
  // Now gs == count_space(str); and s == gs (same computation)
  assert s == gs;

  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := s + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(12,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
12 | {
   | ^

problem_data/p5_7_countWords/program.dfy(11,19): Related location: this is the
    ↪ postcondition that could not be proved
   |
11 |  ensures wordCount == count_space(str) + 1
   | ^^

problem_data/p5_7_countWords/program.dfy(32,17): Error: this invariant could not be
    ↪ proved to be maintained by the loop
 Related message: loop invariant violation
   |
32 | invariant gs == count_space(str[..j])
   | ^^

problem_data/p5_7_countWords/program.dfy(42,2): Error: assertion might not hold
   |
42 | assert s == gs;
   | ^^^^^^


Dafny program verifier finished with 1 verified, 3 errors

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-Co4xPdcw38nukk22PAzzgCf619zHv

Model: gpt-5.1-chat-latest
Created at: 1766051491
Finish reason: stop
Usage: 2311 tokens in, and 405 tokens out

```
{
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    invariant s == count_space(str[..i])
    decreases |str| - i
  {
    if str[i] == ' ' {
      // prove the recursive characterization
      ghost assert count_space(str[..i+1]) == count_space(str[..i]) + 1;
      s := s + 1;
    } else {
      ghost assert count_space(str[..i+1]) == count_space(str[..i]);
    }
    i := i + 1;
  }

  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := s + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(23,12): Error: var expected
   |
23 | ghost assert count_space(str[..i+1]) == count_space(str[..i]) + 1;
   | ^

problem_data/p5_7_countWords/program.dfy(26,12): Error: var expected
   |
26 | ghost assert count_space(str[..i+1]) == count_space(str[..i]);
   | ^

2 parse errors detected in program.dfy

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-Co4xVzZC7mvyvXs4Y3dzk1z66vmYO
Model: gpt-5.1-chat-latest
Created at: 1766051497
Finish reason: stop
Usage: 2603 tokens in, and 204 tokens out

```
{
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    invariant s == count_space(str[..i])
    decreases |str| - i
  {
    if str[i] == ' ' {
      assert count_space(str[..i+1]) == count_space(str[..i]) + 1;
      s := s + 1;
    } else {
      assert count_space(str[..i+1]) == count_space(str[..i]);
    }
    i := i + 1;
  }

  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := s + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(12,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
12 | {
   | ^

problem_data/p5_7_countWords/program.dfy(11,19): Related location: this is the
    ↪ postcondition that could not be proved
   |
11 |  ensures wordCount == count_space(str) + 1
   | ^^

problem_data/p5_7_countWords/program.dfy(22,37): Error: assertion might not hold
   |
22 | assert count_space(str[..i+1]) == count_space(str[..i]) + 1;
   | ^^

problem_data/p5_7_countWords/program.dfy(25,37): Error: assertion might not hold
   |
25 | assert count_space(str[..i+1]) == count_space(str[..i]);
   | ^^


Dafny program verifier finished with 1 verified, 3 errors

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-Co4xYuv5zE2CBb19wIAEBKxUSzCJW
Model: gpt-5.1-chat-latest
Created at: 1766051500
Finish reason: stop
Usage: 3178 tokens in, and 459 tokens out

```
{
  // First loop: count spaces without linking to count_space
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    decreases |str| - i
  {
    if str[i] == ' ' {
      s := s + 1;
    }
    i := i + 1;
  }

  // Second loop (ghost): compute count_space(str)
  ghost var gs := 0;
  ghost var k := 0;
  while k < |str|
    invariant 0 <= k <= |str|
    invariant gs == count_space(str[..k])
    decreases |str| - k
  {
    if str[k] == ' ' {
      gs := gs + 1;
    }
    k := k + 1;
  }

  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := gs + 1;
  }
}
```

Program → GenAI

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

problem_data/p5_7_countWords/program.dfy(44,17): Error: ghost variables such as gs are
    ↪ allowed only in specification contexts. gs was inferred to be ghost based on its
    ↪ declaration or initialization.
   |
44 | wordCount := gs + 1;
   | ^^

1 resolution/type errors detected in program.dfy

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-Co4xfEVfYTgZncsLRwUiDqeRDhxRf
Model: gpt-5.1-chat-latest
Created at: 1766051507
Finish reason: stop
Usage: 3441 tokens in, and 431 tokens out

```
{
  // First loop: compute s = number of spaces (concrete state)
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    decreases |str| - i
  {
    if str[i] == ' ' {
      s := s + 1;
    }
    i := i + 1;
  }

  // Second loop (ghost): prove s == count_space(str)
  ghost var gs := 0;
  ghost var j := 0;
  while j < |str|
    invariant 0 <= j <= |str|
    invariant gs == count_space(str[..j])
    decreases |str| - j
  {
    if str[j] == ' ' {
      gs := gs + 1;
    }
    j := j + 1;
  }

  // Establish equivalence for the postcondition
  assert s == gs;

  // Now use s only (-nonghost)
  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := s + 1;
  }
}
```

Final Program

```
function count_space(str: seq<char>) : int
{
  if |str| == 0 then 0
  else if str[0] == ' ' then 1 + count_space(str[1..])
  else count_space(str[1..])
}

method p5_7_countWords(str: seq<char>) returns (wordCount: int)
       ensures wordCount >= 0
       ensures if |str| == 0 then wordCount == 0 else wordCount >= 1
       ensures wordCount == count_space(str) + 1
{
  // First loop: compute s = number of spaces (concrete state)
  var s := 0;
  var i := 0;
  while i < |str|
    invariant 0 <= i <= |str|
    invariant 0 <= s <= i
    decreases |str| - i
  {
    if str[i] == ' ' {
      s := s + 1;
    }
    i := i + 1;
  }

  // Second loop (ghost): prove s == count_space(str)
  ghost var gs := 0;
  ghost var j := 0;
  while j < |str|
    invariant 0 <= j <= |str|
    invariant gs == count_space(str[..j])
    decreases |str| - j
  {
    if str[j] == ' ' {
      gs := gs + 1;
    }
    j := j + 1;
  }

  // Establish equivalence for the postcondition
  assert s == gs;

  // Now use s only (-nonghost)
  if |str| == 0 {
    wordCount := 0;
  } else {
    wordCount := s + 1;
  }
}
```

Total Token Usage

Input tokens: 15900
Output tokens: 2969
Reasoning tokens: 1216
Sum of ‘total tokens’: 18869

Experiment Timings

Iteration #8 started at 1766051508850, ended at 1766051515385, lasting 6535ms (6.54 seconds)
Overall Experiment started at 1766051468463, ended at 1766051515385, lasting 46922ms (46.92 seconds)
Iteration #4 started at 1766051486186, ended at 1766051493567, lasting 7381ms (7.38 seconds)
Iteration #5 started at 1766051493567, ended at 1766051498949, lasting 5382ms (5.38 seconds)
Iteration #6 started at 1766051498949, ended at 1766051502700, lasting 3751ms (3.75 seconds)
Iteration #7 started at 1766051502700, ended at 1766051508850, lasting 6150ms (6.15 seconds)
Iteration #1 started at 1766051468463, ended at 1766051474217, lasting 5754ms (5.75 seconds)
Iteration #2 started at 1766051474217, ended at 1766051479168, lasting 4951ms (4.95 seconds)
Iteration #3 started at 1766051479168, ended at 1766051486186, lasting 7018ms (7.02 seconds)

Experiment ‘p5_9_coneSurface’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY
Bad responses: 0
Responses containing `assume` : 0
Resolution attempts: 8
Hard fails (resolution): 7
Soft fails (resolution): 0
Verification attempts: 1

Problem Specification

Problem name: p5_9_coneSurface
Natural language statement: Write a method that computes the surface area of a cone with a circular base, given radius and height.
Method signature: p5_9_coneSurface(r: real, h: real) returns (surface_area: real)

Ensures

- `(-r + (surface_area / (3.14159 * r))) * (-r + (surface_area / (3.14159 * r))) == h * h + r * r`

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 computes the surface area of a cone with a circular base, given
    ↪ radius and height.

The signature should be:

method p5_9_coneSurface(r: real, h: real) returns (surface_area: real)

The method should respect the following contract:

ensures (-r + (surface_area / (3.14159 * r))) * (-r + (surface_area / (3.14159 * r))) ==
    ↪ h * h + r * r

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-Co4y9aVJwZZbIT7RURUseiLyZw0A9
Model: gpt-5.1-chat-latest
Created at: 1766051537
Finish reason: stop
Usage: 340 tokens in, and 182 tokens out

```
{
  var slant := sqrt(h * h + r * r);
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(4,15): Error: unresolved identifier: sqrt
  |
4 | var slant := sqrt(h * h + r * r);
  | ^^^^

1 resolution/type errors detected in program.dfy

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-Co4yCRfppRSh5Ha429lIyxIq23Ca0
Model: gpt-5.1-chat-latest
Created at: 1766051540
Finish reason: stop
Usage: 485 tokens in, and 60 tokens out

```
{
  var slant := (h * h + r * r) ** 0.5;
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(4,32): Error: invalid UnaryExpression
  |
4 | var slant := (h * h + r * r) ** 0.5;
  | ^

1 parse errors detected in program.dfy

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-Co4yEa5QGzfL4WrfZ8b2tFGGkoWV3
Model: gpt-5.1-chat-latest
Created at: 1766051542
Finish reason: stop
Usage: 673 tokens in, and 503 tokens out

```
{
  var slant := Math.Sqrt(h * h + r * r);
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(4,15): Error: unresolved identifier: Math
```

```
  |
4 | var slant := Math.Sqrt(h * h + r * r);
  | ^^^^

1 resolution/type errors detected in program.dfy

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-Co4yK2EUWVLx2BHrKW9dIqy9Pmb8M
Model: gpt-5.1-chat-latest
Created at: 1766051548
Finish reason: stop
Usage: 849 tokens in, and 760 tokens out

```
{
  var slant := RealMath.Sqrt(h * h + r * r);
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(4,15): Error: unresolved identifier: RealMath
  |
4 | var slant := RealMath.Sqrt(h * h + r * r);
  | ^^^^^^^^

1 resolution/type errors detected in program.dfy

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-Co4yTlpYi78uzE66UX8OJGXS1Xvs8
Model: gpt-5.1-chat-latest
Created at: 1766051557
Finish reason: stop
Usage: 1034 tokens in, and 695 tokens out

```
{
  var slant := Sqrt(h * h + r * r);
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(4,15): Error: unresolved identifier: Sqrt
  |
4 | var slant := Sqrt(h * h + r * r);
  | ^^^^

1 resolution/type errors detected in program.dfy

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-Co4ybGkJalWptArpvRoCTfOcPuI5x
Model: gpt-5.1-chat-latest
Created at: 1766051565
Finish reason: stop
Usage: 1211 tokens in, and 508 tokens out

```
{
  var slant :| slant * slant == h * h + r * r;
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(2,29): Error: possible division by zero
  |
2 |  ensures (-r + (surface_area / (3.14159 * r))) * (-r + (surface_area / (3.14159 * r))
    ↪ ) == h * h + r * r
  | ^

problem_data/p5_9_coneSurface/program.dfy(3,0): Error: a postcondition could not be
    ↪ proved on this return path
  |
3 | {
  | ^

problem_data/p5_9_coneSurface/program.dfy(2,87): Related location: this is the
    ↪ postcondition that could not be proved
  |
2 |  ensures (-r + (surface_area / (3.14159 * r))) * (-r + (surface_area / (3.14159 * r))
    ↪ ) == h * h + r * r
  | ^^

problem_data/p5_9_coneSurface/program.dfy(4,12): Error: cannot establish the existence of
    ↪  LHS values that satisfy the such-that predicate. Note, no trigger was found for
    ↪ the such-that predicate, which may be the reason the proof failed. To give a
    ↪ trigger explicitly, use the {:trigger} attribute. For more information, see the
    ↪ section on quantifier instantiation rules in the reference manual.
  |
4 | var slant :| slant * slant == h * h + r * r;
  | ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^


Dafny program verifier finished with 0 verified, 3 errors

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-Co4yjutNXGPgJWII08C3oDw3t7uow
Model: gpt-5.1-chat-latest
Created at: 1766051573
Finish reason: stop
Usage: 1945 tokens in, and 534 tokens out

```
{
  if r == 0.0 {
    assert false;
  }
  var slant :| {:trigger slant * slant} slant * slant == h * h + r * r;
  surface_area := 3.14159 * r * (r + slant);
}
```

Program → GenAI

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

problem_data/p5_9_coneSurface/program.dfy(7,15): Error: invalid UnaryExpression
  |
7 | var slant :| {:trigger slant * slant} slant * slant == h * h + r * r;
  | ^

1 parse errors detected in program.dfy

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-Co4yqeaIE0pGxur3zuStMLeUv8fJq
Model: gpt-5.1-chat-latest
Created at: 1766051580
Finish reason: stop
Usage: 1898 tokens in, and 88 tokens out

```
{
  if r == 0.0 {
    assert false;
  }
  {:trigger slant * slant}
  var slant :| slant * slant == h * h + r * r;
  surface_area := 3.14159 * r * (r + slant);
}
```

Final Program

```
method p5_9_coneSurface(r: real, h: real) returns (surface_area: real)
        ensures (-r + (surface_area / (3.14159 * r))) * (-r + (surface_area / (3.14159 * r
            ↪ ))) == h * h + r * r
{
  if r == 0.0 {
    assert false;
  }
  {:trigger slant * slant}
  var slant :| slant * slant == h * h + r * r;
  surface_area := 3.14159 * r * (r + slant);
}
```

Total Token Usage

Input tokens: 8435
Output tokens: 3330
Reasoning tokens: 2816
Sum of ‘total tokens’: 11765

Experiment Timings

Iteration #8 started at 1766051582534, ended at 1766051584932, lasting 2398ms (2.40 seconds)
Overall Experiment started at 1766051538961, ended at 1766051584932, lasting 45971ms (45.97 seconds)
Iteration #4 started at 1766051550671, ended at 1766051559492, lasting 8821ms (8.82 seconds)
Iteration #5 started at 1766051559492, ended at 1766051567676, lasting 8184ms (8.18 seconds)
Iteration #6 started at 1766051567676, ended at 1766051575171, lasting 7495ms (7.50 seconds)
Iteration #7 started at 1766051575171, ended at 1766051582534, lasting 7363ms (7.36 seconds)
Iteration #1 started at 1766051538961, ended at 1766051542050, lasting 3089ms (3.09 seconds)
Iteration #2 started at 1766051542050, ended at 1766051544113, lasting 2063ms (2.06 seconds)
Iteration #3 started at 1766051544113, ended at 1766051550671, lasting 6558ms (6.56 seconds)

Experiment ‘p5_1_a_smallest’ 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_1_a_smallest
Natural language statement: Write a method that returns the smallest of the arguments.
Method signature: p5_1_a_smallest(x: real, y: real, z: real) returns (smallest: real)

Ensures

- `smallest <= x && smallest <= y && smallest <= z`
- `smallest == x || smallest == y || smallest == z`

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 smallest of the arguments.

The signature should be:

method p5_1_a_smallest(x: real, y: real, z: real) returns (smallest: real)

The method should respect the following contract:

ensures smallest <= x && smallest <= y && smallest <= z, ensures smallest == x ||
    ↪ smallest == y || smallest == z

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-Co4tNU0DGMN3CiKCPaS98bqosbuF9
Model: gpt-5.1-chat-latest
Created at: 1766051241
Finish reason: stop
Usage: 298 tokens in, and 129 tokens out

```
{
  var s := x;
  if y < s {
    s := y;
  }
  if z < s {
    s := z;
  }
  smallest := s;
}
```

Final Program

```
method p5_1_a_smallest(x: real, y: real, z: real) returns (smallest: real)
        ensures smallest <= x && smallest <= y && smallest <= z
        ensures smallest == x || smallest == y || smallest == z
{
  var s := x;
  if y < s {
    s := y;
  }
  if z < s {
    s := z;
  }
  smallest := s;
}
```

Total Token Usage

Input tokens: 298
Output tokens: 129
Reasoning tokens: 64
Sum of 'total tokens': 427

Experiment Timings

Overall Experiment started at 1766051242407, ended at 1766051246360, lasting 3953ms (3.95 seconds)
Iteration #1 started at 1766051242410, ended at 1766051246360, lasting 3950ms (3.95 seconds)

Experiment ‘p5_1_b_average’ 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_1_b_average
Natural language statement: Write a method that returns the average of the arguments.
Method signature: p5_1_b_average(x: real, y: real, z: real) returns (average: real)

Ensures

- `average == (x + y + z) / 3.0`

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 average of the arguments.

The signature should be:

method p5_1_b_average(x: real, y: real, z: real) returns (average: real)

The method should respect the following contract:

ensures average == (x + y + z) / 3.0

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-Co4tRL42yJrMujNsaY4WLOTC08DCo
Model: gpt-5.1-chat-latest
Created at: 1766051245
Finish reason: stop
Usage: 274 tokens in, and 37 tokens out

```
{
  average := (x + y + z) / 3.0;
}
```

Final Program

```
method p5_1_b_average(x: real, y: real, z: real) returns (average: real)
        ensures average == (x + y + z) / 3.0
{
  average := (x + y + z) / 3.0;
}
```

Total Token Usage

Input tokens: 274
Output tokens: 37
Reasoning tokens: 0
Sum of ‘total tokens’: 311

Experiment Timings

Overall Experiment started at 1766051247119, ended at 1766051250130, lasting 3011ms (3.01 seconds)
Iteration #1 started at 1766051247119, ended at 1766051250130, lasting 3011ms (3.01 seconds)

Experiment ‘p5_2_a_allTheSame’ 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_2_a_allTheSame
Natural language statement: Write a method returning true if the arguments are all the same.
Method signature: p5_2_a_allTheSame(x: real, y: real, z: real) returns (allSame: bool)

Ensures

- `allSame == (x == y && y == z)`

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 true if the arguments are all the same.

The signature should be:

method p5_2_a_allTheSame(x: real, y: real, z: real) returns (allSame: bool)

The method should respect the following contract:

ensures allSame == (x == y && y == z)

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-Co4tVVYWhIoMzpSHTWmSPBCqh8R0u
Model: gpt-5.1-chat-latest
Created at: 1766051249
Finish reason: stop
Usage: 278 tokens in, and 33 tokens out

```
{
  allSame := x == y && y == z;
}
```

Final Program

```
method p5_2_a_allTheSame(x: real, y: real, z: real) returns (allSame: bool)
        ensures allSame == (x == y && y == z)
{
  allSame := x == y && y == z;
}
```

Total Token Usage

Input tokens: 278
Output tokens: 33
Reasoning tokens: 0
Sum of ‘total tokens’: 311

Experiment Timings

Overall Experiment started at 1766051250868, ended at 1766051253741, lasting 2873ms (2.87 seconds)
Iteration #1 started at 1766051250868, ended at 1766051253741, lasting 2873ms (2.87 seconds)

Experiment ‘p5_2_b_allDifferent’ 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_2_b_allDifferent
Natural language statement: Write a method returning true if the arguments are all different.
Method signature: p5_2_b_allDifferent(x: real, y: real, z: real) returns (allDiff: bool)

Ensures

- `allDiff == (x != y && y != z && x != z)`

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 true if the arguments are all different.

The signature should be:

method p5_2_b_allDifferent(x: real, y: real, z: real) returns (allDiff: bool)

The method should respect the following contract:

ensures allDiff == (x != y && y != z && x != z)

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-Co4tY2V5TClOGIO8WUgzXh9tX3mOE
Model: gpt-5.1-chat-latest
Created at: 1766051252
Finish reason: stop
Usage: 282 tokens in, and 101 tokens out

```
{
  allDiff := x != y && y != z && x != z;
}
```

Final Program

```
method p5_2_b_allDifferent(x: real, y: real, z: real) returns (allDiff: bool)
        ensures allDiff == (x != y && y != z && x != z)
{
  allDiff := x != y && y != z && x != z;
}
```

Total Token Usage

Input tokens: 282
Output tokens: 101
Reasoning tokens: 64
Sum of ‘total tokens’: 383

Experiment Timings

Overall Experiment started at 1766051254376, ended at 1766051257176, lasting 2800ms (2.80 seconds)
Iteration #1 started at 1766051254377, ended at 1766051257176, lasting 2799ms (2.80 seconds)

Experiment ‘p5_2_c_sorted’ 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_2_c_sorted
Natural language statement: Write a method returning true if the arguments are sorted, with the smallest one coming first.
Method signature: p5_2_c_sorted(x: real, y: real, z: real) returns (isSorted: bool)

Ensures

- `isSorted == (x <= y && y <= z)`

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 true if the arguments are sorted, with the smallest one coming
    ↪ first.

The signature should be:

method p5_2_c_sorted(x: real, y: real, z: real) returns (isSorted: bool)

The method should respect the following contract:

ensures isSorted == (x <= y && y <= z)

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-Co4tcAKLT7PRgnyoK9BmvXS1F68jI
Model: gpt-5.1-chat-latest
Created at: 1766051256
Finish reason: stop
Usage: 284 tokens in, and 33 tokens out

```
{
  isSorted := x <= y && y <= z;
}
```

Final Program

```
method p5_2_c_sorted(x: real, y: real, z: real) returns (isSorted: bool)
        ensures isSorted == (x <= y && y <= z)
{
  isSorted := x <= y && y <= z;
}
```

Total Token Usage

Input tokens: 284
Output tokens: 33
Reasoning tokens: 0
Sum of ‘total tokens’: 317

Experiment Timings

Overall Experiment started at 1766051257825, ended at 1766051261248, lasting 3423ms (3.42 seconds)
Iteration #1 started at 1766051257826, ended at 1766051261248, lasting 3422ms (3.42 seconds)

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 <= first < 10`
- `intToSeq(n)[0] == first`

Requires

- `0 <= 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)

Experiment ‘p5_8_scramble’ 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_8_scramble
Natural language statement: Write a method that constructs a scrambled version of string, randomly flipping two characters other than the first and last ones.
Method signature: p5_8_word_scramble(word: string) returns (scrambled: string)

Ensures

- `|word| == |scrambled|`
- `word[0] == scrambled[0]`
- `word[|word| - 1] == scrambled[|scrambled| - 1]`
- `|word| >= 4 ==> exists i, j :: 1 <= i < j <= |word| - 2 && word[i] == scrambled[j] && word[j] == scrambled[i] && forall k:: 0 <= k <= |word| -1 ==> (k!=i && k!=j ==> word[k] == scrambled[k])`

Requires

- `1 <= |word|`

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 constructs a scrambled version of string, randomly flipping two
    ↪ characters other than the first and last ones.

The signature should be:

method p5_8_word_scramble(word: string) returns (scrambled: string)

The method should respect the following contract:

requires 1 <= |word|, ensures |word| == |scrambled|, ensures word[0] == scrambled[0],
    ↪ ensures word[|word| - 1] == scrambled[|scrambled| - 1], ensures |word| >= 4 ==>
    ↪ exists i, j :: 1 <= i < j <= |word| - 2 && word[i] == scrambled[j] && word[j] ==
    ↪ scrambled[i] && forall k:: 0 <= k <= |word| -1 ==> (k!=i && k!=j ==> word[k] ==
    ↪ scrambled[k])

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-Co4xmM5ytrwszEApQB2ZWOFPtkXmM
Model: gpt-5.1-chat-latest
Created at: 1766051514
Finish reason: stop
Usage: 498 tokens in, and 295 tokens out

```
{
  if |word| < 4 {
    scrambled := word;
  } else {
    var i := 1;
    var j := 2;
    var c1 := word[i];
    var c2 := word[j];
    var tmp := word[i := c2];
    scrambled := tmp[j := c1];
  }
}
```

Final Program

```
method p5_8_word_scramble(word: string) returns (scrambled: string)
       requires 1 <= |word|
       ensures |word| == |scrambled|
       ensures word[0] == scrambled[0]
       ensures word[|word| - 1] == scrambled[|scrambled| - 1]
       ensures |word| >= 4 ==> exists i, j :: 1 <= i < j <= |word| - 2 && word[i] ==
           ↪ scrambled[j] && word[j] == scrambled[i] && forall k:: 0 <= k <= |word| -1
           ↪ ==> (k!=i && k!=j ==> word[k] == scrambled[k])
{
  if |word| < 4 {
    scrambled := word;
  } else {
    var i := 1;
    var j := 2;
    var c1 := word[i];
    var c2 := word[j];
    var tmp := word[i := c2];
    scrambled := tmp[j := c1];
  }
}
```

Total Token Usage

Input tokens: 498
Output tokens: 295
Reasoning tokens: 192
Sum of 'total tokens': 793

Experiment Timings

Overall Experiment started at 1766051516108, ended at 1766051521431, lasting 5323ms (5.32 seconds)
Iteration #1 started at 1766051516108, ended at 1766051521431, lasting 5323ms (5.32 seconds)

Experiment ‘p5_9_coneVolume’ 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_9_coneVolume
Natural language statement: Write a method that computes the volume of a cone with a circular base, given radius and height.
Method signature: p5_9_coneVolume(r: real, h: real) returns (volume: real)

Ensures

- `volume == (1.0 / 3.0) * 3.14159 * r * r * h`

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 computes the volume of a cone with a circular base, given radius and
    ↪ height.

The signature should be:

method p5_9_coneVolume(r: real, h: real) returns (volume: real)

The method should respect the following contract:

ensures volume == (1.0 / 3.0) * 3.14159 * r * r * h

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-Co4y5XccEUYXKGKV6y5N9K75mBtNs
Model: gpt-5.1-chat-latest
Created at: 1766051533
Finish reason: stop
Usage: 306 tokens in, and 47 tokens out

```
{
  volume := (1.0 / 3.0) * 3.14159 * r * r * h;
}
```

Final Program

```
method p5_9_coneVolume(r: real, h: real) returns (volume: real)
        ensures volume == (1.0 / 3.0) * 3.14159 * r * r * h
{
  volume := (1.0 / 3.0) * 3.14159 * r * r * h;
}
```

Total Token Usage

Input tokens: 306
Output tokens: 47
Reasoning tokens: 0
Sum of ‘total tokens’: 353

Experiment Timings

Overall Experiment started at 1766051535490, ended at 1766051538308, lasting 2818ms (2.82 seconds)
Iteration #1 started at 1766051535491, ended at 1766051538308, lasting 2817ms (2.82 seconds)

Experiment ‘p5_9_cylinderSurface’ 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_9_cylinderSurface
Natural language statement: Write a method that computes the surface area of a cylinder with a circular base, given radius and height.
Method signature: p5_9_cylinderSurface(r: real, h: real) returns (surface_area: real)

Ensures

- `surface_area == 2.0 * 3.14159 * r * (r + h)`

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 computes the surface area of a cylinder with a circular base, given
    ↪ radius and height.

The signature should be:

method p5_9_cylinderSurface(r: real, h: real) returns (surface_area: real)

The method should respect the following contract:

ensures surface_area == 2.0 * 3.14159 * r * (r + h)

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-Co4y2AZAprfoMdGRfHIba6no2xHGQ
Model: gpt-5.1-chat-latest
Created at: 1766051530
Finish reason: stop
Usage: 302 tokens in, and 43 tokens out

```
{
  surface_area := 2.0 * 3.14159 * r * (r + h);
}
```

Final Program

```
method p5_9_cylinderSurface(r: real, h: real) returns (surface_area: real)
       ensures surface_area == 2.0 * 3.14159 * r * (r + h)
{
  surface_area := 2.0 * 3.14159 * r * (r + h);
}
```

Total Token Usage

Input tokens: 302
Output tokens: 43
Reasoning tokens: 0
Sum of ‘total tokens’: 345

Experiment Timings

Overall Experiment started at 1766051532405, ended at 1766051534870, lasting 2465ms (2.47 seconds)
Iteration #1 started at 1766051532405, ended at 1766051534870, lasting 2465ms (2.47 seconds)

Experiment ‘p5_9_cylinderVolume’ 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_9_cylinderVolume
Natural language statement: Write a method that computes the volume of a cylinder with a circular base, given radius and height.
Method signature: p5_9_cylinderVolume(r: real, h: real) returns (volume: real)

Ensures

- `volume == 3.14159 * r * r * h`

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 computes the volume of a cylinder with a circular base, given radius
    ↪ and height.

The signature should be:

method p5_9_cylinderVolume(r: real, h: real) returns (volume: real)

The method should respect the following contract:

ensures volume == 3.14159 * r * r * h

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-Co4xygHoySaHtQiZxd0ZKVeefmVvP
Model: gpt-5.1-chat-latest
Created at: 1766051526
Finish reason: stop
Usage: 284 tokens in, and 100 tokens out

```
{
  volume := 3.14159 * r * r * h;
}
```

Final Program

```
method p5_9_cylinderVolume(r: real, h: real) returns (volume: real)
       ensures volume == 3.14159 * r * r * h
{
  volume := 3.14159 * r * r * h;
}
```

Total Token Usage

Input tokens: 284
Output tokens: 100
Reasoning tokens: 64
Sum of ‘total tokens’: 384

Experiment Timings

Overall Experiment started at 1766051528530, ended at 1766051531660, lasting 3130ms (3.13 seconds)
Iteration #1 started at 1766051528530, ended at 1766051531660, lasting 3130ms (3.13 seconds)

Experiment 'p5_9_sphereSurface' 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_9_sphereSurface
Natural language statement: Write a method that computes the surface of a sphere given its radius.
Method signature: p5_9_sphereSurface(r: real) returns (surface_area: real)

Ensures

- `surface_area == 4.0 * 3.14159 * r * r`

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 computes the surface of a sphere given its radius.

The signature should be:

method p5_9_sphereSurface(r: real) returns (surface_area: real)

The method should respect the following contract:

ensures surface_area == 4.0 * 3.14159 * r * r

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-Co4xvh9zZi0t2wUo7oOxo9XlTr1Va
Model: gpt-5.1-chat-latest
Created at: 1766051523
Finish reason: stop
Usage: 274 tokens in, and 40 tokens out

```
{
  surface_area := 4.0 * 3.14159 * r * r;
}
```

Final Program

```
method p5_9_sphereSurface(r: real) returns (surface_area: real)
       ensures surface_area == 4.0 * 3.14159 * r * r
{
  surface_area := 4.0 * 3.14159 * r * r;
}
```

Total Token Usage

Input tokens: 274
Output tokens: 40
Reasoning tokens: 0
Sum of ‘total tokens’: 314

Experiment Timings

Overall Experiment started at 1766051525425, ended at 1766051527904, lasting 2479ms (2.48 seconds)
Iteration #1 started at 1766051525425, ended at 1766051527903, lasting 2478ms (2.48 seconds)

Experiment ‘p5_9_sphereVolume’ 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_9_sphereVolume
Natural language statement: Write a method that computes the volume of a sphere given its radius.
Method signature: p5_9_sphereVolume(r: real) returns (volume: real)

Ensures

- `volume == (4.0 / 3.0) * 3.14159 * r * r * r`

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 computes the volume of a sphere given its radius.

The signature should be:

method p5_9_sphereVolume(r: real) returns (volume: real)

The method should respect the following contract:

ensures volume == (4.0 / 3.0) * 3.14159 * r * r * r

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-Co4xs12UPLhFK56WovGXjUq5UFum0
Model: gpt-5.1-chat-latest
Created at: 1766051520
Finish reason: stop
Usage: 286 tokens in, and 47 tokens out

```
{
  volume := (4.0 / 3.0) * 3.14159 * r * r * r;
}
```

Final Program

```
method p5_9_sphereVolume(r: real) returns (volume: real)
        ensures volume == (4.0 / 3.0) * 3.14159 * r * r * r
{
  volume := (4.0 / 3.0) * 3.14159 * r * r * r;
}
```

Total Token Usage

Input tokens: 286
Output tokens: 47
Reasoning tokens: 0
Sum of ‘total tokens’: 333

Experiment Timings

Overall Experiment started at 1766051522066, ended at 1766051524683, lasting 2617ms (2.62 seconds)
Iteration #1 started at 1766051522066, ended at 1766051524683, lasting 2617ms (2.62 seconds)