

Experiment ‘2_8_rectangle_properties’ 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: 2_8_rectangle_properties

Natural language statement: Write a method that takes the lengths of the sides of a rectangle. Then return the area and perimeter of the rectangle and the length of the diagonal (use the Pythagorean theorem).

Method signature: rectangle_properties(length: real, width: real) returns (area: real, perimeter: real, diagonal: real)

Ensures

- $\text{area} == \text{length} * \text{width}$
- $\text{perimeter} == 2.0 * (\text{length} + \text{width})$
- $\text{diagonal} * \text{diagonal} == \text{length} * \text{length} + \text{width} * \text{width}$

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 takes the lengths of the sides of a rectangle. Then return the area
→ and perimeter of the rectangle and the length of the diagonal (use the Pythagorean
→ theorem).

The signature should be:

```
method rectangle_properties(length: real, width: real) returns (area: real, perimeter:  
→ real, diagonal: real)
```

The method should respect the following contract:

```
ensures area == length * width, ensures perimeter == 2.0 * (length + width), ensures  
→ diagonal * diagonal == length * length + width * width
```

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-Co4Pd2JrQ0jX4NXyW62vLmR9l0L3h

Model: gpt-5.1-chat-latest

Created at: 1766049397

Finish reason: stop

Usage: 362 tokens in, and 119 tokens out

```
{  
    area := length * width;  
    perimeter := 2.0 * (length + width);  
    diagonal := sqrt(length * length + width * width);  
}
```

Program → GenAI

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

```
problem_data/2_8_rectangle_properties/program.dfy(8,14): Error: unresolved identifier:  
→ sqrt  
|  
8 | diagonal := sqrt(length * length + width * width);  
| ^~~~
```

```
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-Co4Pg7fOZUsscsEr0zZ4O1KMtnENY

Model: gpt-5.1-chat-latest

Created at: 1766049400

Finish reason: stop

Usage: 491 tokens in, and 379 tokens out

```
{  
    area := length * width;  
    perimeter := 2.0 * (length + width);  
    diagonal :| diagonal * diagonal == length * length + width * width;  
}
```

Program → GenAI

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

problem_data/2_8_rectangle_properties/program.dfy(8,11): 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.

```
|  
8 | diagonal :| diagonal * diagonal == length * length + width * width;  
| ^~
```

Dafny program verifier finished with 0 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-Co4PneNWyV2XiBN4rnn7c0XBjJDR3

Model: gpt-5.1-chat-latest

Created at: 1766049407

Finish reason: stop

Usage: 809 tokens in, and 257 tokens out

```
{
    area := length * width;
    perimeter := 2.0 * (length + width);
    diagonal :| {:trigger diagonal * diagonal} diagonal * diagonal == length * length +
        ↪ width * width;
}
```

Program → GenAI

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

```
problem_data/2_8_rectangle_properties/program.dfy(8,14): Error: invalid UnaryExpression
|
8 | diagonal :| {:trigger diagonal * diagonal} diagonal * diagonal == length * length +
    ↪ width * width;
| ^
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: chatmpl-Co4PrUr0b2UFU1fWFD8rl76cnQcGH

Model: gpt-5.1-chat-latest

Created at: 1766049411

Finish reason: stop

Usage: 937 tokens in, and 129 tokens out

```
{
    area := length * width;
    perimeter := 2.0 * (length + width);
    diagonal :| diagonal * diagonal == length * length + width * width {:trigger diagonal *
        ↪ diagonal};
}
```

Program → GenAI

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

```
problem_data/2_8_rectangle_properties/program.dfy(8,69): Error: invalid AssignStatement
|
8 | diagonal :| diagonal * diagonal == length * length + width * width {:trigger diagonal *
    ↪ * diagonal};
| ^
```

```
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: chatmpl-Co4Ptu41LLbNTp4Nt7eBBu9MBLDZ4

Model: gpt-5.1-chat-latest

Created at: 1766049413

Finish reason: stop

Usage: 1126 tokens in, and 253 tokens out

```
{  
    area := length * width;  
    perimeter := 2.0 * (length + width);  
    diagonal := (length * length + width * width) ** 0.5;  
}
```

Program → GenAI

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

```
problem_data/2_8_rectangle_properties/program.dfy(8,49): Error: invalid UnaryExpression  
|  
8 | diagonal := (length * length + width * width) ** 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: chatmpl-Co4Px5daXt15t5quBYsoUZICgLbMv

Model: gpt-5.1-chat-latest

Created at: 1766049417

Finish reason: stop

Usage: 1306 tokens in, and 196 tokens out

```
{  
    area := length * width;  
    perimeter := 2.0 * (length + width);  
    diagonal :| (diagonal * diagonal == length * length + width * width) {:trigger diagonal  
    ↳ * diagonal};  
}
```

Program → GenAI

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

```
problem_data/2_8_rectangle_properties/program.dfy(8,71): Error: invalid AssignStatement
|
8 | diagonal :| (diagonal * diagonal == length * length + width * width) {:trigger
  |   ↪ diagonal * diagonal};
  |
  |
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-Co4Q1MG6RAG6jeWTf6vnZZ0VcdntC

Model: gpt-5.1-chat-latest

Created at: 1766049421

Finish reason: stop

Usage: 1508 tokens in, and 133 tokens out

```
{
    area := length * width;
    perimeter := 2.0 * (length + width);
    diagonal :| {:trigger (diagonal * diagonal)} (diagonal * diagonal == length * length +
        ↪ width * width);
}
```

Program → GenAI

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

```
problem_data/2_8_rectangle_properties/program.dfy(8,14): Error: invalid UnaryExpression
|
8 | diagonal :| {:trigger (diagonal * diagonal)} (diagonal * diagonal == length * length
  |   ↪ + width * width);
  |
  |
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-Co4Q3WKcrENnJIyBEhjRMT0wI7Z6w

Model: gpt-5.1-chat-latest

Created at: 1766049423

Finish reason: stop

Usage: 1709 tokens in, and 257 tokens out

```
{  
    area := length * width;  
    perimeter := 2.0 * (length + width);  
    diagonal {::trigger diagonal * diagonal} ::| diagonal * diagonal == length * length +  
        ↪ width * width;  
}
```

Final Program

```
method rectangle_properties(length: real, width: real) returns (area: real, perimeter:  
    ↪ real, diagonal: real)  
    ensures area == length * width  
    ensures perimeter == 2.0 * (length + width)  
    ensures diagonal * diagonal == length * length + width * width  
{  
    area := length * width;  
    perimeter := 2.0 * (length + width);  
    diagonal {::trigger diagonal * diagonal} ::| diagonal * diagonal == length * length +  
        ↪ width * width;  
}
```

Total Token Usage

Input tokens: 8248

Output tokens: 1723

Reasoning tokens: 1216

Sum of ‘total tokens’: 9971

Experiment Timings

Iteration #8 started at 1766049425635, ended at 1766049429361, lasting 3726ms (3.73 seconds)
Overall Experiment started at 1766049399676, ended at 1766049429361, lasting 29685ms (29.69 seconds)
Iteration #4 started at 1766049412784, ended at 1766049415694, lasting 2910ms (2.91 seconds)
Iteration #5 started at 1766049415694, ended at 1766049419728, lasting 4034ms (4.03 seconds)
Iteration #6 started at 1766049419728, ended at 1766049423008, lasting 3280ms (3.28 seconds)
Iteration #7 started at 1766049423008, ended at 1766049425635, lasting 2627ms (2.63 seconds)
Iteration #1 started at 1766049399676, ended at 1766049402482, lasting 2806ms (2.81 seconds)
Iteration #2 started at 1766049402482, ended at 1766049408686, lasting 6204ms (6.20 seconds)
Iteration #3 started at 1766049408686, ended at 1766049412784, lasting 4098ms (4.10 seconds)

Experiment ‘P_2_2_Letter_perimeter’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY

Bad responses: 0

Responses containing assume : 0

Resolution attempts: 8

Hard fails (resolution): 4

Soft fails (resolution): 0

Verification attempts: 4

Problem Specification

Problem name: P_2_2_Letter_perimeter

Natural language statement: Write a method that computes and returns the perimeter of a letter-size (8.5 x 11 inches) sheet of paper and the length of its diagonal.

Method signature: letter_perimeter() returns (perimeter:real, diagonal:real)

Ensures

- $\text{perimeter} == (8.5 * 25.4) * 2.0 + 11.0 * 25.4 * 2.0$
- $\text{diagonal} * \text{diagonal} == (8.5 * 25.4) * (8.5 * 25.4) + 11.0 * 25.4 * 11.0 * 25.4$

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 and returns the perimeter of a letter-size (8.5 x 11 inches)
sheet of paper and the length of its diagonal.

The signature should be:

```
method letter_perimeter() returns (perimeter:real, diagonal:real)
```

The method should respect the following contract:

```
ensures perimeter == (8.5 * 25.4) * 2.0 + 11.0 * 25.4 * 2.0, ensures diagonal * diagonal  
== (8.5 * 25.4) * (8.5 * 25.4) + 11.0 * 25.4 * 11.0 * 25.4
```

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4O55TRkMNwdHpB425gbImaYLLpg

Model: gpt-5.1-chat-latest

Created at: 1766049301

Finish reason: stop

Usage: 428 tokens in, and 227 tokens out

```
{  
    var width: real := 8.5 * 25.4;  
    var height: real := 11.0 * 25.4;  
  
    perimeter := width * 2.0 + height * 2.0;  
  
    var diagSquared: real := width * width + height * height;  
    diagonal := sqrt(diagSquared);  
}
```

Program → GenAI

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

```
problem_data/P_2_2_Letter_perimeter/program.dfy(11,14): Error: unresolved identifier:  
    ↪ sqrt  
    |  
11 | diagonal := sqrt(diagSquared);
```

```
| ^^^^
```

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: chatmpl-Co4OBaPx3nt7ET6YDlAI6IyPnNId

Model: gpt-5.1-chat-latest

Created at: 1766049307

Finish reason: stop

Usage: 566 tokens in, and 993 tokens out

```
{
  var width: real := 8.5 * 25.4;
  var height: real := 11.0 * 25.4;

  perimeter := width * 2.0 + height * 2.0;

  var diagSquared: real := width * width + height * height;
  diagonal := diagSquared;
}
```

Program → GenAI

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

```
problem_data/P_2_2_Letter_perimeter/program.dfy(4,0): Error: a postcondition could not be
  ↪ proved on this return path
|
4 | {
| ^
| 

problem_data/P_2_2_Letter_perimeter/program.dfy(3,29): Related location: this is the
  ↪ postcondition that could not be proved
|
3 | ensures diagonal * diagonal == (8.5 * 25.4) * (8.5 * 25.4) + 11.0 * 25.4 * 11.0 *
  ↪ 25.4
| ^^
```

Dafny program verifier finished with 0 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-Co4OMsHBES8qvCoQ162tGL18hLFrS

Model: gpt-5.1-chat-latest

Created at: 1766049318

Finish reason: stop

Usage: 971 tokens in, and 934 tokens out

```
{  
    var width: real := 8.5 * 25.4;  
    var height: real := 11.0 * 25.4;  
  
    perimeter := width * 2.0 + height * 2.0;  
  
    var diagSquared: real := width * width + height * height;  
    diagonal :| diagonal * diagonal == diagSquared;  
}
```

Program → GenAI

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

```
problem_data/P_2_2_Letter_perimeter/program.dfy(11,11): 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.  
 |  
11 | diagonal :| diagonal * diagonal == diagSquared;  
 | ^~
```

Dafny program verifier finished with 0 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-Co4OYkUSVIaLpl92x7XMxv0GCOzkx

Model: gpt-5.1-chat-latest

Created at: 1766049330

Finish reason: stop

Usage: 1234 tokens in, and 172 tokens out

```
{  
    var width: real := 8.5 * 25.4;  
    var height: real := 11.0 * 25.4;
```

```

perimeter := width * 2.0 + height * 2.0;

var diagSquared: real := width * width + height * height;
diagonal :| {:trigger diagonal * diagonal} diagonal * diagonal == diagSquared;
}

```

Program → GenAI

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

```

problem_data/P_2_2_Letter_perimeter/program.dfy(11,14): Error: invalid UnaryExpression
|
11 | diagonal :| {:trigger diagonal * diagonal} diagonal * diagonal == diagSquared;
| ^
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-Co4ObToDtCoJ5Mp7Sb4ukYqxtIIBP

Model: gpt-5.1-chat-latest

Created at: 1766049333

Finish reason: stop

Usage: 1403 tokens in, and 172 tokens out

```
{
    var width: real := 8.5 * 25.4;
    var height: real := 11.0 * 25.4;

    perimeter := width * 2.0 + height * 2.0;

    var diagSquared: real := width * width + height * height;
    diagonal :| diagonal * diagonal == diagSquared {:trigger diagonal * diagonal};
}
```

Program → GenAI

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

```

problem_data/P_2_2_Letter_perimeter/program.dfy(11,49): Error: invalid AssignStatement
|
11 | diagonal :| diagonal * diagonal == diagSquared {:trigger diagonal * diagonal};
| ^
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-Co4OeVGvztNuD9Vi60rXNgm5FtkRW

Model: gpt-5.1-chat-latest

Created at: 1766049336

Finish reason: stop

Usage: 1633 tokens in, and 237 tokens out

```
{  
    var width: real := 8.5 * 25.4;  
    var height: real := 11.0 * 25.4;  
  
    perimeter := width * 2.0 + height * 2.0;  
  
    var diagSquared: real := width * width + height * height;  
    diagonal :| ({:trigger diagonal * diagonal} diagonal * diagonal == diagSquared);  
}
```

Program → GenAI

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

```
problem_data/P_2_2_Letter_perimeter/program.dfy(11,15): Error: invalid ParensExpression  
|  
11 | diagonal :| ({:trigger diagonal * diagonal} diagonal * diagonal == diagSquared);  
| ^  
  
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-Co4Ohs8QnUesaw71npSvdnySlZJv

Model: gpt-5.1-chat-latest

Created at: 1766049339

Finish reason: stop

Usage: 1871 tokens in, and 767 tokens out

```
{  
    var width: real := 8.5 * 25.4;  
    var height: real := 11.0 * 25.4;
```

```

perimeter := width * 2.0 + height * 2.0;

var diagSquared: real := width * width + height * height;

// Use a witness value directly equal to sqrt(diagSquared)^2
// diagonal * diagonal must equal diagSquared, so we choose diagonal = diagSquared /
    ↪ diagonal
// To avoid sqrt, we select the positive root by defining diagonal as diagSquared / (
    ↪ width + height)
// and then scale it to satisfy the equation exactly.

diagonal := diagSquared / (width + height);
diagonal := diagonal * (width + height) / diagonal;
}

```

Program → GenAI

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

```

problem_data/P_2_2_Letter_perimeter/program.dfy(4,0): Error: a postcondition could not be
    ↪ proved on this return path
|
4 | {
| ^
|
problem_data/P_2_2_Letter_perimeter/program.dfy(3,29): Related location: this is the
    ↪ postcondition that could not be proved
|
3 | ensures diagonal * diagonal == (8.5 * 25.4) * (8.5 * 25.4) + 11.0 * 25.4 * 11.0 *
    ↪ 25.4
| ^


```

Dafny program verifier finished with 0 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-Co4OrDFMGGgL7Gbj9encXnMZKssJP

Model: gpt-5.1-chat-latest

Created at: 1766049349

Finish reason: stop

Usage: 2362 tokens in, and 296 tokens out

```
{
    var width: real := 8.5 * 25.4;
```

```

var height: real := 11.0 * 25.4;

perimeter := width * 2.0 + height * 2.0;

var diagSquared: real := width * width + height * height;
diagonal :| (diagonal * diagonal == diagSquared);
}

```

Final Program

```

method letter_perimeter() returns (perimeter:real, diagonal:real)
    ensures perimeter == (8.5 * 25.4) * 2.0 + 11.0 * 25.4 * 2.0
    ensures diagonal * diagonal == (8.5 * 25.4) * (8.5 * 25.4) + 11.0 * 25.4 * 11.0 *
        ↪ 25.4
{
    var width: real := 8.5 * 25.4;
    var height: real := 11.0 * 25.4;

    perimeter := width * 2.0 + height * 2.0;

    var diagSquared: real := width * width + height * height;
    diagonal :| (diagonal * diagonal == diagSquared);
}

```

Total Token Usage

Input tokens: 10468
Output tokens: 3798
Reasoning tokens: 2880
Sum of ‘total tokens’: 14266

Experiment Timings

Iteration #8 started at 1766049351233, ended at 1766049356789, lasting 5556ms (5.56 seconds)
Overall Experiment started at 1766049302252, ended at 1766049356789, lasting 54537ms (54.54 seconds)
Iteration #4 started at 1766049332097, ended at 1766049335483, lasting 3386ms (3.39 seconds)
Iteration #5 started at 1766049335484, ended at 1766049338341, lasting 2857ms (2.86 seconds)
Iteration #6 started at 1766049338341, ended at 1766049341549, lasting 3208ms (3.21 seconds)
Iteration #7 started at 1766049341549, ended at 1766049351233, lasting 9684ms (9.68 seconds)
Iteration #1 started at 1766049302252, ended at 1766049308659, lasting 6407ms (6.41 seconds)
Iteration #2 started at 1766049308660, ended at 1766049320687, lasting 12027ms (12.03 seconds)
Iteration #3 started at 1766049320687, ended at 1766049332097, lasting 11410ms (11.41 seconds)

Experiment ‘2_10_hybrid_car_cost’ 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: 2_10_hybrid_car_cost

Natural language statement: Write a method that helps a person decide whether to buy a hybrid car. Your methods’s inputs should be: the cost of a new car, the estimated miles driven per year, the estimated gas price, the efficiency in miles per gallon, the estimated resale value after 5 years. Compute the total cost of owning the car for five years.

Method signature: hybrid_car_cost(new_car_cost: real, miles_per_year: real, gas_price: real, mpg: real, resale_value: real) returns (total_cost: real)

Ensures

- `total_cost == new_car_cost + (miles_per_year / mpg) * gas_price * 5.0 - resale_value`

Requires

- `0.0 < mpg`

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 helps a person decide whether to buy a hybrid car. Your methods's
→ inputs should be: the cost of a new car, the estimated miles driven per year, the
→ estimated gas price, the efficiency in miles per gallon, the estimated resale
→ value after 5 years. Compute the total cost of owning the car for five years.

The signature should be:

```
method hybrid_car_cost(new_car_cost: real, miles_per_year: real, gas_price: real, mpg:  
→ real, resale_value: real) returns (total_cost: real)
```

The method should respect the following contract:

```
requires 0.0 < mpg, ensures total_cost == new_car_cost + (miles_per_year / mpg) *  
→ gas_price * 5.0 - resale_value
```

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-Co4Q8JUiOJmPGuWuoFWtM0tpxOQ7

Model: gpt-5.1-chat-latest

Created at: 1766049428

Finish reason: stop

Usage: 452 tokens in, and 113 tokens out

```
{  
    total_cost := new_car_cost + (miles_per_year / mpg) * gas_price * 5.0 - resale_value;  
}
```

Final Program

```
method hybrid_car_cost(new_car_cost: real, miles_per_year: real, gas_price: real, mpg:  
→ real, resale_value: real) returns (total_cost: real)  
    requires 0.0 < mpg  
    ensures total_cost == new_car_cost + (miles_per_year / mpg) * gas_price * 5.0 -  
        → resale_value  
{  
    total_cost := new_car_cost + (miles_per_year / mpg) * gas_price * 5.0 - resale_value;  
}
```

Total Token Usage

Input tokens: 452

Output tokens: 113

Reasoning tokens: 64

Sum of ‘total tokens’: 565

Experiment Timings

Overall Experiment started at 1766049430066, ended at 1766049433441, lasting 3375ms (3.38 seconds)

Iteration #1 started at 1766049430067, ended at 1766049433441, lasting 3374ms (3.37 seconds)

Experiment ‘2_4_6_max’ 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: 2_4_6_max

Natural language statement: Write a method that takes two integers and then returns the maximum (the larger of the two).

Method signature: max(x: int, y: int) returns (result:int)

Ensures

- `result == x || result == y`
- `if x >= y then result == x else result == y`

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 takes two integers and then returns the maximum (the larger of the ↪ two).

The signature should be:

```
method max(x: int, y: int) returns (result:int)
```

The method should respect the following contract:

```
ensures result == x || result == y, ensures if x >= y then result == x else result == y
```

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4PKVaGzlMAthLdbwSCJhnA3VIV1

Model: gpt-5.1-chat-latest

Created at: 1766049378

Finish reason: stop

Usage: 284 tokens in, and 111 tokens out

```
{
    if x >= y {
        result := x;
    } else {
        result := y;
    }
}
```

Final Program

```
method max(x: int, y: int) returns (result:int)
    ensures result == x || result == y
    ensures if x >= y then result == x else result == y
{
    if x >= y {
        result := x;
    } else {
        result := y;
    }
}
```

}

Total Token Usage

Input tokens: 284

Output tokens: 111

Reasoning tokens: 64

Sum of ‘total tokens’: 395

Experiment Timings

Overall Experiment started at 1766049379764, ended at 1766049382773, lasting 3009ms (3.01 seconds)

Iteration #1 started at 1766049379765, ended at 1766049382773, lasting 3008ms (3.01 seconds)

Experiment ‘2_4_7_min’ 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: 2_4_7_min

Natural language statement: Write a method that takes two integers and then returns the minimum (the smaller of the two).

Method signature: min(x: int, y: int) returns (result:int)

Ensures

- result == x || result == y
- if x <= y then result == x else result == y

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 takes two integers and then returns the minimum (the smaller of the ↪ two).

The signature should be:

```
method min(x: int, y: int) returns (result:int)
```

The method should respect the following contract:

```
ensures result == x || result == y, ensures if x <= y then result == x else result == y
```

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4PNPXWZTgsadbxy5Kn6koWPMfZ

Model: gpt-5.1-chat-latest

Created at: 1766049381

Finish reason: stop

Usage: 284 tokens in, and 111 tokens out

```
{  
    if x <= y {  
        result := x;  
    } else {  
        result := y;  
    }  
}
```

Final Program

```
method min(x: int, y: int) returns (result:int)  
    ensures result == x || result == y  
    ensures if x <= y then result == x else result == y  
{  
    if x <= y {  
        result := x;  
    } else {  
        result := y;
```

}

Total Token Usage

Input tokens: 284

Output tokens: 111

Reasoning tokens: 64

Sum of ‘total tokens’: 395

Experiment Timings

Overall Experiment started at 1766049383421, ended at 1766049386638, lasting 3217ms (3.22 seconds)

Iteration #1 started at 1766049383421, ended at 1766049386638, lasting 3217ms (3.22 seconds)

Experiment ‘2_6_convert’ 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: 2_6_convert

Natural language statement: Write a method that takes a measurement in meters and then converts it to miles, feet, and inches.

Method signature: convert(x: real) returns (miles: real, feet: real, inches: real)

Ensures

- $x / 0.000621371 == \text{miles}$
- $x / 3.28084 == \text{feet}$
- $x / 39.3701 == \text{inches}$

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 takes a measurement in meters and then converts it to miles, feet,
↪ and inches.

The signature should be:

```
method convert(x: real) returns (miles: real, feet: real, inches: real)
```

The method should respect the following contract:

```
ensures x / 0.000621371 == miles, ensures x / 3.28084 == feet, ensures x / 39.3701 ==  
↪ inches
```

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-Co4PRfsmbq1BnJSanMHjkeqduG3qs

Model: gpt-5.1-chat-latest

Created at: 1766049385

Finish reason: stop

Usage: 322 tokens in, and 121 tokens out

```
{  
    miles := x / 0.000621371;  
    feet := x / 3.28084;  
    inches := x / 39.3701;  
}
```

Final Program

```
method convert(x: real) returns (miles: real, feet: real, inches: real)  
    ensures x / 0.000621371 == miles  
    ensures x / 3.28084 == feet  
    ensures x / 39.3701 == inches  
{  
    miles := x / 0.000621371;  
    feet := x / 3.28084;  
    inches := x / 39.3701;  
}
```

Total Token Usage

Input tokens: 322

Output tokens: 121

Reasoning tokens: 64

Sum of ‘total tokens’: 443

Experiment Timings

Overall Experiment started at 1766049387283, ended at 1766049392751, lasting 5468ms (5.47 seconds)

Iteration #1 started at 1766049387283, ended at 1766049392750, lasting 5467ms (5.47 seconds)

Experiment ‘2_7_circle_properties’ 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: 2_7_circle_properties

Natural language statement: Write a method that takes a radius and then returns the area and circumference of a circle with that radius and the volume and surface area of a sphere with that radius.

Method signature: circle_properties(radius: real) returns (circumference: real, area: real, surface_area: real, volume: real)

Ensures

- circumference == $2.0 * 3.14159 * \text{radius}$
- area == $3.14159 * \text{radius} * \text{radius}$
- surface_area == $4.0 * 3.14159 * \text{radius} * \text{radius}$
- volume == $(4.0 / 3.0) * 3.14159 * \text{radius} * \text{radius} * \text{radius}$

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 takes a radius and then returns the area and circumference of a circle with that radius and the volume and surface area of a sphere with that radius.

The signature should be:

```
method circle_properties(radius: real) returns (circumference: real, area: real,  
    ↪ surface_area: real, volume: real)
```

The method should respect the following contract:

```
ensures circumference == 2.0 * 3.14159 * radius, ensures area == 3.14159 * radius *  
    ↪ radius, ensures surface_area == 4.0 * 3.14159 * radius * radius, ensures volume ==  
    ↪ (4.0 / 3.0) * 3.14159 * radius * radius * radius
```

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4PYaliEOeUDWuAc90IHeSydYUWk

Model: gpt-5.1-chat-latest

Created at: 1766049392

Finish reason: stop

Usage: 440 tokens in, and 290 tokens out

```
{  
    circumference := 2.0 * 3.14159 * radius;  
    area := 3.14159 * radius * radius;  
    surface_area := 4.0 * 3.14159 * radius * radius;  
    volume := (4.0 / 3.0) * 3.14159 * radius * radius * radius;  
}
```

Final Program

```
method circle_properties(radius: real) returns (circumference: real, area: real,  
    ↪ surface_area: real, volume: real)  
    ensures circumference == 2.0 * 3.14159 * radius  
    ensures area == 3.14159 * radius * radius  
    ensures surface_area == 4.0 * 3.14159 * radius * radius
```

```
    ensures volume == (4.0 / 3.0) * 3.14159 * radius * radius * radius
{
    circumference := 2.0 * 3.14159 * radius;
    area := 3.14159 * radius * radius;
    surface_area := 4.0 * 3.14159 * radius * radius;
    volume := (4.0 / 3.0) * 3.14159 * radius * radius * radius;
}
```

Total Token Usage

Input tokens: 440

Output tokens: 290

Reasoning tokens: 192

Sum of ‘total tokens’: 730

Experiment Timings

Overall Experiment started at 1766049393392, ended at 1766049398916, lasting 5524ms (5.52 seconds)

Iteration #1 started at 1766049393392, ended at 1766049398916, lasting 5524ms (5.52 seconds)

Experiment ‘P_2_1_Letter_Dimensions’ 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: P_2_1_Letter_Dimensions

Natural language statement: Write a method that computes and returns the dimensions of a letter-size (8.5 x 11 inches) sheet of paper in millimeters. There are 25.4 millimeters per inch. Use constants and comments in your method.

Method signature: letter_dimensions() returns (width:real, height:real)

Ensures

- width == 8.5 * 25.4
- height == 11.0 * 25.4

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 and returns the dimensions of a letter-size (8.5 x 11 inches
→) sheet of paper in millimeters. There are 25.4 millimeters per inch. Use
→ constants and comments in your method.

The signature should be:

```
method letter_dimensions() returns (width:real, height:real)
```

The method should respect the following contract:

```
ensures width == 8.5 * 25.4, ensures height == 11.0 * 25.4
```

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4NzuBmIlgdrU4FHYSHbc7v2kXiU

Model: gpt-5.1-chat-latest

Created at: 1766049295

Finish reason: stop

Usage: 346 tokens in, and 226 tokens out

```
{
    // millimeters per inch
    var INCH_MM := 25.4;
    // letter size in inches
    var WIDTH_IN := 8.5;
    var HEIGHT_IN := 11.0;
    // compute dimensions in millimeters
    width := WIDTH_IN * INCH_MM;
    height := HEIGHT_IN * INCH_MM;
}
```

Final Program

```
method letter_dimensions() returns (width:real, height:real)
    ensures width == 8.5 * 25.4
    ensures height == 11.0 * 25.4
{
```

```
// millimeters per inch
var INCH_MM := 25.4;
// letter size in inches
var WIDTH_IN := 8.5;
var HEIGHT_IN := 11.0;
// compute dimensions in millimeters
width := WIDTH_IN * INCH_MM;
height := HEIGHT_IN * INCH_MM;
}
```

Total Token Usage

Input tokens: 346

Output tokens: 226

Reasoning tokens: 128

Sum of ‘total tokens’: 572

Experiment Timings

Overall Experiment started at 1766049295994, ended at 1766049301597, lasting 5603ms (5.60 seconds)

Iteration #1 started at 1766049295995, ended at 1766049301597, lasting 5602ms (5.60 seconds)

Experiment ‘P_2_3_powerFunctions’ 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: P_2_3_powerFunctions

Natural language statement: Write a method that takes a number and returns its square, cube, and fourth power.

Method signature: power_functions(x: int) returns (sq: int, cube : int, fourth: int)

Ensures

- `sq == x * x`
- `cube == x * x * x`
- `fourth == x * x * x * x`

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 takes a number and returns its square, cube, and fourth power.

The signature should be:

```
method power_functions(x: int) returns (sq: int, cube : int, fourth: int)
```

The method should respect the following contract:

```
ensures sq == x * x, ensures cube == x * x * x, ensures fourth == x * x * x * x
```

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-Co4OxJ0KZHNUX9G3qfJ2SYClpYVgI

Model: gpt-5.1-chat-latest

Created at: 1766049355

Finish reason: stop

Usage: 302 tokens in, and 50 tokens out

```
{
    sq := x * x;
    cube := x * x * x;
    fourth := x * x * x * x;
}
```

Final Program

```
method power_functions(x: int) returns (sq: int, cube : int, fourth: int)
    ensures sq == x * x
    ensures cube == x * x * x
    ensures fourth == x * x * x * x
{
    sq := x * x;
    cube := x * x * x;
    fourth := x * x * x * x;
}
```

Total Token Usage

Input tokens: 302

Output tokens: 50

Reasoning tokens: 0

Sum of ‘total tokens’: 352

Experiment Timings

Overall Experiment started at 1766049357586, ended at 1766049360178, lasting 2592ms (2.59 seconds)

Iteration #1 started at 1766049357586, ended at 1766049360178, lasting 2592ms (2.59 seconds)

Experiment ‘P_2_4_1_sum’ 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: P_2_4_1_sum

Natural language statement: Write a method that takes two integers and then returns the sum.

Method signature: sum(x: int, y: int) returns (result:int)

Ensures

- `result == x + y`

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 takes two integers and then returns the sum.

The signature should be:

```
method sum(x: int, y: int) returns (result:int)
```

The method should respect the following contract:

```
ensures result == x + y
```

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-Co4P2PgMlzPdv97jY9XLHRD0rRnz5

Model: gpt-5.1-chat-latest

Created at: 1766049360

Finish reason: stop

Usage: 240 tokens in, and 28 tokens out

```
{  
    result := x + y;  
}
```

Final Program

```
method sum(x: int, y: int) returns (result:int)  
    ensures result == x + y  
{  
    result := x + y;  
}
```

Total Token Usage

Input tokens: 240

Output tokens: 28

Reasoning tokens: 0

Sum of ‘total tokens’: 268

Experiment Timings

Overall Experiment started at 1766049360845, ended at 1766049364085, lasting 3240ms (3.24 seconds)
Iteration #1 started at 1766049360845, ended at 1766049364085, lasting 3240ms (3.24 seconds)

Experiment ‘P_2_4_2_difference’ 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: P_2_4_2_difference

Natural language statement: Write a method that takes two integers and then returns the difference.

Method signature: difference(x: int, y: int) returns (result:int)

Ensures

- `result == x - y`

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 takes two integers and then returns the difference.

The signature should be:

```
method difference(x: int, y: int) returns (result:int)
```

The method should respect the following contract:

```
ensures result == x - y
```

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-Co4P5w77Kw9r97sNwzn9qVR3oefkB

Model: gpt-5.1-chat-latest

Created at: 1766049363

Finish reason: stop

Usage: 240 tokens in, and 28 tokens out

```
{  
    result := x - y;  
}
```

Final Program

```
method difference(x: int, y: int) returns (result:int)  
    ensures result == x - y  
{  
    result := x - y;  
}
```

Total Token Usage

Input tokens: 240

Output tokens: 28

Reasoning tokens: 0

Sum of ‘total tokens’: 268

Experiment Timings

Overall Experiment started at 1766049364838, ended at 1766049367953, lasting 3115ms (3.12 seconds)
Iteration #1 started at 1766049364839, ended at 1766049367953, lasting 3114ms (3.11 seconds)

Experiment ‘P_2_4_3_product’ 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: P_2_4_3_product

Natural language statement: Write a method that takes two integers and then returns the product.

Method signature: product(x: int, y: int) returns (result:int)

Ensures

- `result == x * y`

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 takes two integers and then returns the product.

The signature should be:

```
method product(x: int, y: int) returns (result:int)
```

The method should respect the following contract:

```
ensures result == x * y
```

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-Co4P9sOlTK1vMpzms2OEHOxKAxtsP

Model: gpt-5.1-chat-latest

Created at: 1766049367

Finish reason: stop

Usage: 240 tokens in, and 92 tokens out

```
{  
    result := x * y;  
}
```

Final Program

```
method product(x: int, y: int) returns (result:int)  
    ensures result == x * y  
{  
    result := x * y;  
}
```

Total Token Usage

Input tokens: 240

Output tokens: 92

Reasoning tokens: 64

Sum of ‘total tokens’: 332

Experiment Timings

Overall Experiment started at 1766049368593, ended at 1766049371893, lasting 3300ms (3.30 seconds)
Iteration #1 started at 1766049368593, ended at 1766049371893, lasting 3300ms (3.30 seconds)

Experiment ‘P_2_4_4_avarage’ 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: P_2_4_4_avarage

Natural language statement: Write a method that takes two integers and then returns the average.

Method signature: average(x: int, y: int) returns (result:real)

Ensures

- `result == (x + y) as real / 2.0`

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 takes two integers and then returns the average.

The signature should be:

```
method average(x: int, y: int) returns (result:real)
```

The method should respect the following contract:

```
ensures result == (x + y) as real / 2.0
```

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-Co4PCLULvG5PnH4NMhSIFUT9vXfVq

Model: gpt-5.1-chat-latest

Created at: 1766049370

Finish reason: stop

Usage: 260 tokens in, and 37 tokens out

```
{  
    result := (x + y) as real / 2.0;  
}
```

Final Program

```
method average(x: int, y: int) returns (result:real)  
    ensures result == (x + y) as real / 2.0  
{  
    result := (x + y) as real / 2.0;  
}
```

Total Token Usage

Input tokens: 260

Output tokens: 37

Reasoning tokens: 0

Sum of ‘total tokens’: 297

Experiment Timings

Overall Experiment started at 1766049372553, ended at 1766049375075, lasting 2522ms (2.52 seconds)
Iteration #1 started at 1766049372554, ended at 1766049375075, lasting 2521ms (2.52 seconds)

Experiment ‘P_2_4_5_distance’ 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: P_2_4_5_distance

Natural language statement: Write a method that takes two integers and then returns the distance (absolute value of the difference).

Method signature: distance(x: int, y: int) returns (result:int)

Ensures

- `result >= 0`
- `if x >= y then result == x - y else result == y - x`

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 takes two integers and then returns the distance (absolute value of
→ the difference).

The signature should be:

```
method distance(x: int, y: int) returns (result:int)
```

The method should respect the following contract:

```
ensures result >= 0, ensures if x >= y then result == x - y else result == y - x
```

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4PFsClCBUnodP9wSb6MJKO7eXeN

Model: gpt-5.1-chat-latest

Created at: 1766049373

Finish reason: stop

Usage: 286 tokens in, and 115 tokens out

```
{
    if x >= y {
        result := x - y;
    } else {
        result := y - x;
    }
}
```

Final Program

```
method distance(x: int, y: int) returns (result:int)
    ensures result >= 0
    ensures if x >= y then result == x - y else result == y - x
{
    if x >= y {
        result := x - y;
    } else {
        result := y - x;
```

}

Total Token Usage

Input tokens: 286

Output tokens: 115

Reasoning tokens: 64

Sum of ‘total tokens’: 401

Experiment Timings

Overall Experiment started at 1766049375725, ended at 1766049379008, lasting 3283ms (3.28 seconds)

Iteration #1 started at 1766049375725, ended at 1766049379008, lasting 3283ms (3.28 seconds)

