

# Experiment ‘P\_3\_10\_temperature\_altitude’ 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\_3\_10\_temperature\_altitude

**Natural language statement:** Write a method that takes a temperature value and the altitude in meters. Return whether water is liquid, solid, or gaseous at the given temperature. The boiling point of water drops by about one degree centigrade for every 300 meters of altitude.

**Method signature:** P\_3\_10\_temperature\_altitude (t: real, a: nat) returns (s: string)

### Ensures

- if  $t < 0.0$  then  $s == \text{"solid"}$  else if  $t == 0.0$  then  $s == \text{"solid or liquid"}$  else if  $0.0 < t < 100.0 - (a \text{ as real} / 300.0)$  then  $s == \text{"liquid"}$  else if  $t == 100.0 - (a \text{ as real} / 300.0)$  then  $s == \text{"liquid or gaseous"}$  else  $s == \text{"gaseous"}$

### Requires

- $t \geq -273.15 \ \&& a \leq 10000$

## 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 temperature value and the altitude in meters. Return whether  
→ water is liquid, solid, or gaseous at the given temperature. The boiling point of  
→ water drops by about one degree centigrade for every 300 meters of altitude.

The signature should be:

```
method P_3_10_temperature_altitude (t: real, a: nat) returns (s: string)
```

The method should respect the following contract:

```
requires t >= -273.15 && a <= 10000, ensures if t < 0.0 then s == "solid" else if t ==  
→ 0.0 then s == "solid or liquid" else if 0.0 < t < 100.0 - (a as real / 300.0) then  
→ s == "liquid" else if t == 100.0 - (a as real / 300.0) then s == "liquid or  
→ gaseous" else s == "gaseous"
```

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-Co4T5Xw9lMurLBMFh9X2Jy33Gp3bi

**Model:** gpt-5.1-chat-latest

**Created at:** 1766049611

**Finish reason:** stop

**Usage:** 542 tokens in, and 329 tokens out

```
{  
    var bp := 100.0 - (a as real / 300.0);  
    if t < 0.0 {  
        s := "solid";  
    } else if t == 0.0 {  
        s := "solid or liquid";  
    } else if t < bp {  
        s := "liquid";  
    } else if t == bp {  
        s := "liquid or gaseous";  
    } else {  
        s := "gaseous";  
    }  
}
```

## Final Program

```
method P_3_10_temperature_altitude (t: real, a: nat) returns (s: string)
    requires t >= -273.15 && a <= 10000
    ensures if t < 0.0 then s == "solid" else if t == 0.0 then s == "solid or liquid"
        ↪ else if 0.0 < t < 100.0 - (a as real / 300.0) then s == "liquid" else if t
        ↪ == 100.0 - (a as real / 300.0) then s == "liquid or gaseous" else s == "
        ↪ gaseous"
{
    var bp := 100.0 - (a as real / 300.0);
    if t < 0.0 {
        s := "solid";
    } else if t == 0.0 {
        s := "solid or liquid";
    } else if t < bp {
        s := "liquid";
    } else if t == bp {
        s := "liquid or gaseous";
    } else {
        s := "gaseous";
    }
}
```

## Total Token Usage

**Input tokens:** 542

**Output tokens:** 329

**Reasoning tokens:** 192

**Sum of ‘total tokens’:** 871

## Experiment Timings

**Overall Experiment** started at 1766049613031, ended at 1766049618118, lasting 5087ms (5.09 seconds)

**Iteration #1** started at 1766049613031, ended at 1766049618118, lasting 5087ms (5.09 seconds)



# Experiment ‘P\_3\_1\_classify\_int’ 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\_3\_1\_classify\_int

**Natural language statement:** Write a method that takes an integer and returns whether it is negative, zero, or positive.

**Method signature:** P\_3\_1\_classify\_int (x:int) returns (s: string)

### Ensures

- $x == 0 \implies s == \text{"zero"}$
- $x > 0 \implies s == \text{"positive"}$
- $x < 0 \implies s == \text{"negative"}$

## 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 an integer and returns whether it is negative, zero, or  
→ positive.

The signature should be:

```
method P_3_1_classify_int (x:int) returns (s: string)
```

The method should respect the following contract:

```
ensures x == 0 ==> s == "zero", ensures x > 0 ==> s == "positive", ensures x < 0 ==> s ==  
→ "negative"
```

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-Co4S0XKxA1nr3ffR69VSb1ljVrpHS

Model: gpt-5.1-chat-latest

Created at: 1766049544

Finish reason: stop

Usage: 312 tokens in, and 70 tokens out

```
{  
    if x == 0 {  
        s := "zero";  
    } else if x > 0 {  
        s := "positive";  
    } else {  
        s := "negative";  
    }  
}
```

### Final Program

```
method P_3_1_classify_int (x:int) returns (s: string)  
    ensures x == 0 ==> s == "zero"  
    ensures x > 0 ==> s == "positive"  
    ensures x < 0 ==> s == "negative"  
{
```

```
if x == 0 {  
    s := "zero";  
} else if x > 0 {  
    s := "positive";  
} else {  
    s := "negative";  
}  
}
```

## Total Token Usage

**Input tokens:** 312

**Output tokens:** 70

**Reasoning tokens:** 0

**Sum of ‘total tokens’:** 382

## Experiment Timings

**Overall Experiment** started at 1766049545918, ended at 1766049549055, lasting 3137ms (3.14 seconds)

**Iteration #1** started at 1766049545921, ended at 1766049549055, lasting 3134ms (3.13 seconds)



# Experiment ‘P\_3\_2\_classify\_real’ Results

December 18, 2025

**Experiment outcome:** SUCCESS

**Bad responses:** 0

**Responses containing assume :** 0

**Resolution attempts:** 2

**Hard fails (resolution):** 1

**Soft fails (resolution):** 0

**Verification attempts:** 1

## Problem Specification

**Problem name:** P\_3\_2\_classify\_real

**Natural language statement:** Write a method that takes a floating point number and returns whether it is negative, zero, or positive. Add "small" if the absolute value of the number is less than 1, or "large" if it exceeds 1,000,000.

**Method signature:** P\_3\_2\_classify\_real (x:real) returns (s: string)

### Ensures

- $x == 0.0 \implies s == \text{"zero, small"}$
- $1.0 > x > 0.0 \implies s == \text{"positive, small"}$
- $1.0 \leq x < 1000000.0 \implies s == \text{"positive"}$
- $1000000.0 \leq x \implies s == \text{"positive, large"}$
- $0.0 > x > -1.0 \implies s == \text{"negative, small"}$
- $-1.0 \geq x > -1000000.0 \implies s == \text{"negative"}$
- $-1000000.0 \geq x \implies s == \text{"negative, large"}$

## 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 floating point number and returns whether it is negative,  
↳ zero, or positive. Add "small" if the absolute value of the number is less than 1,  
↳ or "large" if it exceeds 1,000,000.

The signature should be:

```
method P_3_2_classify_real (x:real) returns (s: string)
```

The method should respect the following contract:

```
ensures x == 0.0 ==> s == "zero, small", ensures 1.0 > x > 0.0 ==> s == "positive, small  
↳ ", ensures 1.0 <= x < 1000000.0 ==> s == "positive", ensures 1000000.0 <= x ==> s  
↳ == "positive, large", ensures 0.0 > x > -1.0 ==> s == "negative, small", ensures  
↳ -1.0 >= x > -1000000.0 ==> s == "negative", ensures -1000000.0 >= x ==> s == "  
↳ negative, large"
```

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-Co4S4PPipSMR3cI8b4TTuZMdM0tj0

**Model:** gpt-5.1-chat-latest

**Created at:** 1766049548

**Finish reason:** stop

**Usage:** 574 tokens in, and 155 tokens out

```
{  
    if x == 0.0 then  
        s := "zero, small"  
    else if x > 0.0 then  
        if x < 1.0 then  
            s := "positive, small"  
        else if x < 1000000.0 then  
            s := "positive"  
        else  
            s := "positive, large"  
    else  
        if x > -1.0 then  
            s := "negative, small"  
        else if x > -1000000.0 then  
            s := "negative"
```

```

    else
        s := "negative, large"
}

```

## Program → GenAI

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

```

problem_data/P_3_2_classify_real/program.dfy(10,14): Error: lbrace expected
|
10 | if x == 0.0 then
| ^
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-Co4S9PN2KCVm6OS5cTwymg8zZ0V2i

**Model:** gpt-5.1-chat-latest

**Created at:** 1766049553

**Finish reason:** stop

**Usage:** 691 tokens in, and 180 tokens out

```
{
    if x == 0.0 {
        s := "zero, small";
    } else if x > 0.0 {
        if x < 1.0 {
            s := "positive, small";
        } else if x < 1000000.0 {
            s := "positive";
        } else {
            s := "positive, large";
        }
    } else {
        if x > -1.0 {
            s := "negative, small";
        } else if x > -1000000.0 {
            s := "negative";
        } else {
            s := "negative, large";
        }
    }
}
```

## Final Program

```
method P_3_2_classify_real (x:real) returns (s: string)
    ensures x == 0.0 ==> s == "zero, small"
    ensures 1.0 > x > 0.0 ==> s == "positive, small"
    ensures 1.0 <= x < 1000000.0 ==> s == "positive"
    ensures 1000000.0 <= x ==> s == "positive, large"
    ensures 0.0 > x > -1.0 ==> s == "negative, small"
    ensures -1.0 >= x > -1000000.0 ==> s == "negative"
    ensures -1000000.0 >= x ==> s == "negative, large"
{
    if x == 0.0 {
        s := "zero, small";
    } else if x > 0.0 {
        if x < 1.0 {
            s := "positive, small";
        } else if x < 1000000.0 {
            s := "positive";
        } else {
            s := "positive, large";
        }
    } else {
        if x > -1.0 {
            s := "negative, small";
        } else if x > -1000000.0 {
            s := "negative";
        } else {
            s := "negative, large";
        }
    }
}
```

## Total Token Usage

Input tokens: 1265  
Output tokens: 335  
Reasoning tokens: 0  
Sum of 'total tokens': 1600

## Experiment Timings

Overall Experiment started at 1766049549817, ended at 1766049559160, lasting 9343ms (9.34 seconds)  
Iteration #1 started at 1766049549818, ended at 1766049555292, lasting 5474ms (5.47 seconds)  
Iteration #2 started at 1766049555292, ended at 1766049559160, lasting 3868ms (3.87 seconds)



# Experiment ‘P\_3\_3\_NumberOfDigits’ Results

December 18, 2025

**Experiment outcome:** SUCCESS

**Bad responses:** 0

**Responses containing assume :** 0

**Resolution attempts:** 4

**Hard fails (resolution):** 0

**Soft fails (resolution):** 0

**Verification attempts:** 4

## Problem Specification

**Problem name:** P\_3\_3\_NumberOfDigits

**Natural language statement:** Write a method that takes an integer and returns how many digits the number has, by checking whether the number is 10, 100, and so on. (Assume that all integers are less than ten billion.) If the number is negative, first multiply it with -1.

**Method signature:** P\_3\_3\_NumberOfDigits (i:int) returns (n: nat)

### Ensures

- $i \geq 0 \implies n = \text{digits}(i)$
- $i < 0 \implies n = \text{digits}(-i)$

### Requires

- $10000000000 > i > -10000000000$

## Functional Code Given

```
function digits (n:nat): nat
{
    if 0 <= n <= 9 then 1
    else 1 + digits (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 that takes an integer and returns how many digits the number has, by  
→ checking whether the number is 10, 100, and so on. (Assume that all integers are  
→ less than ten billion.) If the number is negative, first multiply it with -1.

The signature should be:

```
method P_3_3_NumberOfDigits (i:int) returns (n: nat)
```

The method should respect the following contract:

```
requires 10000000000 > i > -10000000000, ensures i >= 0 ==> n == digits(i), ensures i < 0  
→ ==> n == digits(-i)
```

The contract uses the following dafny code:

```
function digits (n:nat): nat  
{  
    if 0 <= n <= 9 then 1  
    else 1 + digits (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-Co4SELCLKpj1IRciaRs2BV2FxYqq9

**Model:** gpt-5.1-chat-latest

**Created at:** 1766049558

**Finish reason:** stop

**Usage:** 518 tokens in, and 278 tokens out

```
{  
    var j := if i < 0 then -i else i;  
    if j < 10 {  
        n := 1;  
    } else if j < 100 {  
        n := 2;  
    } else if j < 1000 {  
        n := 3;  
    }
```

```

} else if j < 10000 {
    n := 4;
} else if j < 100000 {
    n := 5;
} else if j < 1000000 {
    n := 6;
} else if j < 10000000 {
    n := 7;
} else if j < 100000000 {
    n := 8;
} else if j < 1000000000 {
    n := 9;
} else {
    n := 10;
}
}

```

## Program → GenAI

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

```

problem_data/P_3_3_NumberOfDigits/program.dfy(11,0): Error: a postcondition could not be
    ↪ proved on this return path
|
11 | {
| ^
|


problem_data/P_3_3_NumberOfDigits/program.dfy(9,22): Related location: this is the
    ↪ postcondition that could not be proved
|
9 | ensures i >= 0 ==> n == digits(i)
| ^


problem_data/P_3_3_NumberOfDigits/program.dfy(11,0): Error: a postcondition could not be
    ↪ proved on this return path
|
11 | {
| ^


problem_data/P_3_3_NumberOfDigits/program.dfy(10,21): Related location: this is the
    ↪ postcondition that could not be proved
|
10 | ensures i < 0 ==> n == digits(-i)
| ^


```

Dafny program verifier finished with 2 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: chatmpl-Co4SJ7xeT3l1VSpjvg4FFZ7HFjxbA

Model: gpt-5.1-chat-latest

Created at: 1766049563

Finish reason: stop

Usage: 1032 tokens in, and 212 tokens out

```
{  
    var j := if i < 0 then -i else i;  
    n := 0;  
    var k := j;  
    while k > 0  
        invariant k >= 0  
        invariant n >= 0  
        invariant if i >= 0 then (n + digits(k)) == digits(i)  
                  else (n + digits(k)) == digits(-i)  
    {  
        n := n + 1;  
        k := k / 10;  
    }  
    if j == 0 {  
        n := 1;  
    }  
}
```

## Program → GenAI

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

```
problem_data/P_3_3_NumberOfDigits/program.dfy(11,0): Error: a postcondition could not be  
    ↪ proved on this return path  
    |  
11 | {  
| ^  
  
problem_data/P_3_3_NumberOfDigits/program.dfy(9,22): Related location: this is the  
    ↪ postcondition that could not be proved  
    |  
9 | ensures i >= 0 ==> n == digits(i)  
| ^^  
  
problem_data/P_3_3_NumberOfDigits/program.dfy(11,0): Error: a postcondition could not be  
    ↪ proved on this return path  
    |  
11 | {  
| ^  
  
problem_data/P_3_3_NumberOfDigits/program.dfy(10,21): Related location: this is the  
    ↪ postcondition that could not be proved  
    |
```

```

10 | ensures i < 0 ==> n == digits(-i)
| ^^

problem_data/P_3_3_NumberOfDigits/program.dfy(18,44): Error: this invariant could not be
  ↵ proved to be maintained by the loop
Related message: loop invariant violation
|
18 | invariant if i >= 0 then (n + digits(k) == digits(i))
| ^^

problem_data/P_3_3_NumberOfDigits/program.dfy(19,34): Error: this invariant could not be
  ↵ proved to be maintained by the loop
Related message: loop invariant violation
|
19 | else (n + digits(k) == digits(-i))
| ^^

Dafny program verifier finished with 2 verified, 4 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-Co4SO8SaJw7QX8SgfLsbY4iawibOd

**Model:** gpt-5.1-chat-latest

**Created at:** 1766049568

**Finish reason:** stop

**Usage:** 1705 tokens in, and 408 tokens out

```
{
  var j:nat := if i < 0 then -i else i;
  if j < 10 {
    n := 1;
  } else if j < 100 {
    n := 2;
  } else if j < 1000 {
    n := 3;
  } else if j < 10000 {
    n := 4;
  } else if j < 100000 {
    n := 5;
  } else if j < 1000000 {
    n := 6;
  } else if j < 10000000 {
    n := 7;
  } else if j < 100000000 {
    n := 8;
  }
}
```

```
    } else if j < 10000000000 {  
        n := 9;  
    } else {  
        n := 10;  
    }  
}
```

## Program → GenAI

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

```
problem_data/P_3_3_NumberOfDigits/program.dfy(11,0): Error: a postcondition could not be
  ↪ proved on this return path
  |
11 | {
  | ^
  |
problem_data/P_3_3_NumberOfDigits/program.dfy(9,22): Related location: this is the
  ↪ postcondition that could not be proved
  |
9 |   ensures i >= 0 ==> n == digits(i)
  | ^

problem_data/P_3_3_NumberOfDigits/program.dfy(11,0): Error: a postcondition could not be
  ↪ proved on this return path
  |
11 | {
  | ^
  |
problem_data/P_3_3_NumberOfDigits/program.dfy(10,21): Related location: this is the
  ↪ postcondition that could not be proved
  |
10 |   ensures i < 0 ==> n == digits(-i)
  | ^
```

Dafny program verifier finished with 2 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:** chatmpl-Co4SWvhrgp9DBPDEqE3uZWtjW7pan

**Model:** gpt-5.1-chat-latest

Created at: 1766049576

**Finish reason:** stop

**Usage:** 2065 tokens in, and 152 tokens out

```
{
    var j:nat := if i < 0 then -i else i;
    var k:nat := j;
    n := 0;

    if k == 0 {
        n := 1;
        return;
    }

    while k > 0
        invariant k > 0 ==> digits(k) + n == digits(j)
        invariant k == 0 ==> n == digits(j)
        invariant n >= 0
        invariant k <= j
    {
        n := n + 1;
        k := k / 10;
    }
}
}
```

## Final Program

```
function digits (n:nat): nat
{
    if 0 <= n <= 9 then 1
    else 1 + digits (n/10)
}

method P_3_3_NumberOfDigits (i:int) returns (n: nat)
    requires 10000000000 > i > -10000000000
    ensures i >= 0 ==> n == digits(i)
    ensures i < 0 ==> n == digits(-i)
{
    var j:nat := if i < 0 then -i else i;
    var k:nat := j;
    n := 0;

    if k == 0 {
        n := 1;
        return;
    }

    while k > 0
        invariant k > 0 ==> digits(k) + n == digits(j)
        invariant k == 0 ==> n == digits(j)
        invariant n >= 0
        invariant k <= j
    {
        n := n + 1;
    }
}
```

```
    k := k / 10;  
}  
}
```

## Total Token Usage

**Input tokens:** 5320

**Output tokens:** 1050

**Reasoning tokens:** 320

**Sum of ‘total tokens’:** 6370

## Experiment Timings

**Overall Experiment** started at 1766049559813, ended at 1766049582378, lasting 22565ms (22.57 seconds)

**Iteration #4** started at 1766049577936, ended at 1766049582378, lasting 4442ms (4.44 seconds)

**Iteration #1** started at 1766049559814, ended at 1766049565512, lasting 5698ms (5.70 seconds)

**Iteration #2** started at 1766049565512, ended at 1766049570305, lasting 4793ms (4.79 seconds)

**Iteration #3** started at 1766049570305, ended at 1766049577936, lasting 7631ms (7.63 seconds)



# Experiment ‘P\_3\_4\_Classification’ 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\_3\_4\_Classification

**Natural language statement:** Write a method that takes three numbers and returns "all the same" if they are all the same, "all different" if they are all different, and "neither" otherwise.

**Method signature:** P\_3\_4\_Classification (i:int, j: int, k: int) returns (s: string)

### Ensures

- if i == j == k then s == "all the same" else if i != j && j!=k && i!=k then s == "all different" else s== "neither"

## 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 three numbers and returns "all the same" if they are all the  
→ same, "all different" if they are all different, and "neither" otherwise.

The signature should be:

```
method P_3_4_Classification (i:int, j: int, k: int) returns (s: string)
```

The method should respect the following contract:

```
ensures if i == j == k then s == "all the same" else if i != j && j!=k && i!=k then s ==  
→ "all different" else s== "neither"
```

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-Co4SbufU6j92JeBxsuQnKglRbIb2

Model: gpt-5.1-chat-latest

Created at: 1766049581

Finish reason: stop

Usage: 382 tokens in, and 84 tokens out

```
{  
    if i == j && j == k {  
        s := "all the same";  
    } else if i != j && j != k && i != k {  
        s := "all different";  
    } else {  
        s := "neither";  
    }  
}
```

### Final Program

```
method P_3_4_Classification (i:int, j: int, k: int) returns (s: string)  
    ensures if i == j == k then s == "all the same" else if i != j && j!=k && i!=k  
        → then s == "all different" else s== "neither"  
{  
    if i == j && j == k {
```

```
s := "all the same";
} else if i != j && j != k && i != k {
    s := "all different";
} else {
    s := "neither";
}
}
```

## Total Token Usage

**Input tokens:** 382

**Output tokens:** 84

**Reasoning tokens:** 0

**Sum of ‘total tokens’:** 466

## Experiment Timings

**Overall Experiment** started at 1766049583094, ended at 1766049586006, lasting 2912ms (2.91 seconds)

**Iteration #1** started at 1766049583094, ended at 1766049586006, lasting 2912ms (2.91 seconds)



# Experiment ‘P\_3\_5\_Order’ 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\_3\_5\_Order

**Natural language statement:** Write a method that takes three numbers and returns "increasing" if they are in increasing order, "decreasing" if they are in decreasing order, and "neither" otherwise. Here, "increasing" means "strictly increasing", with each value larger than its predecessor. The sequence 3 4 4 would not be considered increasing.

**Method signature:** P\_3\_5\_Order (i:int, j: int, k: int) returns (s: string)

### Ensures

- if i < j < k then s == "increasing" else if i > j && j > k then s == "decreasing" else s== "neither"

## 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 three numbers and returns "increasing" if they are in  
→ increasing order, "decreasing" if they are in decreasing order, and "neither"  
→ otherwise. Here, "increasing" means "strictly increasing", with each value larger  
→ than its predecessor. The sequence 3 4 4 would not be considered increasing.

The signature should be:

```
method P_3_5_Order (i:int, j: int, k: int) returns (s: string)
```

The method should respect the following contract:

```
ensures if i < j < k then s == "increasing" else if i > j && j > k then s == "decreasing"  
→ else s== "neither"
```

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-Co4Se0sHq29NHzjjJL3SUYHURJEG4

Model: gpt-5.1-chat-latest

Created at: 1766049584

Finish reason: stop

Usage: 438 tokens in, and 143 tokens out

```
{  
    if i < j && j < k {  
        s := "increasing";  
    } else if i > j && j > k {  
        s := "decreasing";  
    } else {  
        s := "neither";  
    }  
}
```

### Final Program

```
method P_3_5_Order (i:int, j: int, k: int) returns (s: string)  
ensures if i < j < k then s == "increasing" else if i > j && j > k then s == "  
→ decreasing" else s== "neither"
```

```
{  
    if i < j && j < k {  
        s := "increasing";  
    } else if i > j && j > k {  
        s := "decreasing";  
    } else {  
        s := "neither";  
    }  
}
```

## Total Token Usage

**Input tokens:** 438

**Output tokens:** 143

**Reasoning tokens:** 64

**Sum of ‘total tokens’:** 581

## Experiment Timings

**Overall Experiment** started at 1766049586651, ended at 1766049590221, lasting 3570ms (3.57 seconds)

**Iteration #1** started at 1766049586651, ended at 1766049590221, lasting 3570ms (3.57 seconds)



# Experiment ‘P\_3\_6\_ParametrizedOrder’ 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\_3\_6\_ParametrizedOrder

**Natural language statement:** Write a method that takes three numbers and returns "increasing" if they are in increasing order, "decreasing" if they are in decreasing order, and "neither" otherwise. A Boolean parameter triggers the method to be "strict" or "lenient". In lenient mode, the sequence 3 4 4 is increasing and the sequence 4 4 4 is both increasing and decreasing.

**Method signature:** P\_3\_6\_ParametrizedOrder (i:int, j: int, k: int, strict: bool ) returns (s: string)

### Ensures

- if strict then if i < j < k then s == "increasing" else if i > j && j > k then s == "decreasing" else s == "neither" else if i <= j <= k then if i == j == k then s == "increasing and decreasing" else s == "increasing" else if i >= j && j >= k then s == "decreasing" else s == "neither"

## 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 three numbers and returns "increasing" if they are in  
→ increasing order, "decreasing" if they are in decreasing order, and "neither"  
→ otherwise. A Boolean parameter triggers the method to be "strict" or "lenient". In  
→ lenient mode, the sequence 3 4 4 is increasing and the sequence 4 4 4 is both  
→ increasing and decreasing.

The signature should be:

```
method P_3_6_ParametrizedOrder (i:int, j: int, k: int, strict: bool ) returns (s: string)
```

The method should respect the following contract:

ensures if strict then if i < j < k then s == "increasing" else if i > j && j > k then s  
→ == "decreasing" else s == "neither" else if i <= j <= k then if i == j == k then s  
→ == "increasing and decreasing" else s == "increasing" else if i >= j && j >= k  
→ then s == "decreasing" else s == "neither"

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-Co4SjnWKJCnzRbeNYpKI59vfCSr0B

Model: gpt-5.1-chat-latest

Created at: 1766049589

Finish reason: stop

Usage: 588 tokens in, and 245 tokens out

```
{
  if strict {
    if i < j && j < k {
      s := "increasing";
    } else if i > j && j > k {
      s := "decreasing";
    } else {
      s := "neither";
    }
  } else {
    if i <= j && j <= k {
      if i == j && j == k {
        s := "increasing and decreasing";
      } else {
```

```

        s := "increasing";
    }
} else if i >= j && j >= k {
    s := "decreasing";
} else {
    s := "neither";
}
}
}

```

## Final Program

```

method P_3_6_ParametrizedOrder (i:int, j: int, k: int, strict: bool ) returns (s: string)
    ensures if strict then if i < j < k then s == "increasing" else if i > j && j > k
        then s == "decreasing" else s == "neither" else if i <= j <= k then if i ==
            j == k then s == "increasing and decreasing" else s == "increasing" else
            if i >= j && j >= k then s == "decreasing" else s == "neither"
{
    if strict {
        if i < j && j < k {
            s := "increasing";
        } else if i > j && j > k {
            s := "decreasing";
        } else {
            s := "neither";
        }
    } else {
        if i <= j && j <= k {
            if i == j && j == k {
                s := "increasing and decreasing";
            } else {
                s := "increasing";
            }
        } else if i >= j && j >= k {
            s := "decreasing";
        } else {
            s := "neither";
        }
    }
}

```

## Total Token Usage

**Input tokens:** 588  
**Output tokens:** 245  
**Reasoning tokens:** 64  
**Sum of ‘total tokens’:** 833

## Experiment Timings

**Overall Experiment** started at 1766049590872, ended at 1766049595896, lasting 5024ms (5.02 seconds)  
**Iteration #1** started at 1766049590872, ended at 1766049595896, lasting 5024ms (5.02 seconds)



# Experiment ‘P\_3\_7\_Order’ 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\_3\_7\_Order

**Natural language statement:** Write a method that takes three integers and returns "in order" if they are sorted in ascending or descending order, or "not in order" otherwise.

**Method signature:** P\_3\_7\_Order (i:int, j: int, k: int) returns (s: string)

### Ensures

- if (i < j < k) || (i > j > k) then s == "in order" else s == "not in order"

## 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 three integers and returns "in order" if they are sorted in  
→ ascending or descending order, or "not in order" otherwise.

The signature should be:

```
method P_3_7_Order (i:int, j: int, k: int) returns (s: string)
```

The method should respect the following contract:

```
ensures if (i < j < k) || (i > j > k) then s == "in order" else s == "not in order"
```

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-Co4SovRRcuhFYhAaNEakyzMt0GkJq

**Model:** gpt-5.1-chat-latest

**Created at:** 1766049594

**Finish reason:** stop

**Usage:** 346 tokens in, and 134 tokens out

```
{
    if ((i < j && j < k) || (i > j && j > k)) {
        s := "in order";
    } else {
        s := "not in order";
    }
}
```

## Final Program

```
method P_3_7_Order (i:int, j: int, k: int) returns (s: string)
    ensures if (i < j < k) || (i > j > k) then s == "in order" else s == "not in order"
    → "
{
    if ((i < j && j < k) || (i > j && j > k)) {
        s := "in order";
    } else {
        s := "not in order";
    }
}
```

}

## Total Token Usage

**Input tokens:** 346

**Output tokens:** 134

**Reasoning tokens:** 64

**Sum of ‘total tokens’:** 480

## Experiment Timings

**Overall Experiment** started at 1766049596599, ended at 1766049600527, lasting 3928ms (3.93 seconds)

**Iteration #1** started at 1766049596600, ended at 1766049600527, lasting 3927ms (3.93 seconds)



# Experiment ‘P\_3\_8\_TwoPairs’ 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\_3\_8\_TwoPairs

**Natural language statement:** Write a method that takes four integers and returns “two pairs” if the input consists of two matching pairs (in some order) and “not two pairs” otherwise.

**Method signature:** P\_3\_8\_TwoPairs (i:int, j: int, k: int, l: int) returns (s: string)

### Ensures

- if (i==j && k==l) || (i==k && j==l) || (i==l && j==k) then s == "two pairs" else s == "not two pairs"

## 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 four integers and returns "two pairs" if the input consists of  
↳ two matching pairs (in some order) and "not two pairs" otherwise.

The signature should be:

```
method P_3_8_TwoPairs (i:int, j: int, k: int, l: int) returns (s: string)
```

The method should respect the following contract:

```
ensures if (i==j && k==l) || (i==k && j==l) || (i==l && j==k) then s == "two pairs" else  
↳ s == "not two pairs"
```

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-Co4StP0ZTe6uuwSTIEfW3mHTydDps

Model: gpt-5.1-chat-latest

Created at: 1766049599

Finish reason: stop

Usage: 392 tokens in, and 80 tokens out

```
{  
    if ((i == j && k == l) || (i == k && j == l) || (i == l && j == k)) {  
        s := "two pairs";  
    } else {  
        s := "not two pairs";  
    }  
}
```

## Final Program

```
method P_3_8_TwoPairs (i:int, j: int, k: int, l: int) returns (s: string)  
    ensures if (i==j && k==l) || (i==k && j==l) || (i==l && j==k) then s == "two pairs"  
        ↳ " else s == "not two pairs"  
{  
    if ((i == j && k == l) || (i == k && j == l) || (i == l && j == k)) {  
        s := "two pairs";  
    } else {  
    }
```

```
    s := "not two pairs";
}
}
```

## Total Token Usage

**Input tokens:** 392

**Output tokens:** 80

**Reasoning tokens:** 0

**Sum of ‘total tokens’:** 472

## Experiment Timings

**Overall Experiment** started at 1766049601160, ended at 1766049603979, lasting 2819ms (2.82 seconds)

**Iteration #1** started at 1766049601161, ended at 1766049603979, lasting 2818ms (2.82 seconds)



# Experiment ‘P\_3\_9\_temperature’ 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\_3\_9\_temperature

**Natural language statement:** Write a method that takes a temperature value and a letter C for Celsius or F for Fahrenheit. Return whether water is liquid, solid, or gaseous at the given temperature at sea level.

**Method signature:** P\_3\_9\_temperature (t: real, c: char) returns (s: string)

### Ensures

- if c == 'C' then if t < 0.0 then s == "solid" else if t == 0.0 then s == "solid or liquid" else if 0.0 < t < 100.0 then s == "liquid" else if t == 100.0 then s == "liquid or gaseous" else s == "gaseous" else if t < 32.0 then s == "solid" else if t == 32.0 then s == "solid or liquid" else if 32.0 < t < 212.0 then s == "liquid" else if t == 212.0 then s == "liquid or gaseous" else s == "gaseous"

### Requires

- (c == 'C' && t >= -273.15) || (c == 'F' && t >= -459.67)
- c == 'C' || c == 'F'

## 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 temperature value and a letter C for Celsius or F for  
↳ Fahrenheit. Return whether water is liquid, solid, or gaseous at the given  
↳ temperature at sea level.

The signature should be:

```
method P_3_9_temperature (t: real, c: char) returns (s: string)
```

The method should respect the following contract:

```
requires (c == 'C' && t >= -273.15) || (c == 'F' && t >= -459.67), requires c == 'C' || c  
↳ == 'F', ensures if c == 'C' then if t < 0.0 then s == "solid" else if t == 0.0  
↳ then s == "solid or liquid" else if 0.0 < t < 100.0 then s == "liquid" else if t  
↳ == 100.0 then s == "liquid or gaseous" else s == "gaseous" else if t < 32.0 then s  
↳ == "solid" else if t == 32.0 then s == "solid or liquid" else if 32.0 < t < 212.0  
↳ then s == "liquid" else if t == 212.0 then s == "liquid or gaseous" else s == "  
↳ gaseous"
```

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-Co4SwiayBMFCZeXMbxJsqnCsXv33T

Model: gpt-5.1-chat-latest

Created at: 1766049602

Finish reason: stop

Usage: 690 tokens in, and 499 tokens out

```
{  
    if c == 'C' {  
        if t < 0.0 {  
            s := "solid";  
        } else if t == 0.0 {  
            s := "solid or liquid";  
        } else if t < 100.0 {  
            s := "liquid";  
        } else if t == 100.0 {  
            s := "liquid or gaseous";  
        } else {  
            s := "gaseous";  
        }  
    }  
}
```

```

} else {
    if t < 32.0 {
        s := "solid";
    } else if t == 32.0 {
        s := "solid or liquid";
    } else if t < 212.0 {
        s := "liquid";
    } else if t == 212.0 {
        s := "liquid or gaseous";
    } else {
        s := "gaseous";
    }
}
}

```

## Final Program

```

method P_3_9_temperature (t: real, c: char) returns (s: string)
    requires (c == 'C' && t >= -273.15) || (c == 'F' && t >= -459.67)
    requires c == 'C' || c == 'F'
    ensures if c == 'C' then if t < 0.0 then s == "solid" else if t == 0.0 then s == "
        ↪ solid or liquid" else if 0.0 < t < 100.0 then s == "liquid" else if t ==
        ↪ 100.0 then s == "liquid or gaseous" else s == "gaseous" else if t < 32.0
        ↪ then s == "solid" else if t == 32.0 then s == "solid or liquid" else if
        ↪ 32.0 < t < 212.0 then s == "liquid" else if t == 212.0 then s == "liquid or
        ↪ gaseous" else s == "gaseous"
{
    if c == 'C' {
        if t < 0.0 {
            s := "solid";
        } else if t == 0.0 {
            s := "solid or liquid";
        } else if t < 100.0 {
            s := "liquid";
        } else if t == 100.0 {
            s := "liquid or gaseous";
        } else {
            s := "gaseous";
        }
    } else {
        if t < 32.0 {
            s := "solid";
        } else if t == 32.0 {
            s := "solid or liquid";
        } else if t < 212.0 {
            s := "liquid";
        } else if t == 212.0 {
            s := "liquid or gaseous";
        } else {
            s := "gaseous";
        }
    }
}

```

```
    }  
}  
}
```

## Total Token Usage

**Input tokens:** 690

**Output tokens:** 499

**Reasoning tokens:** 256

**Sum of ‘total tokens’:** 1189

## Experiment Timings

**Overall Experiment** started at 1766049604627, ended at 1766049612281, lasting 7654ms (7.65 seconds)

**Iteration #1** started at 1766049604627, ended at 1766049612281, lasting 7654ms (7.65 seconds)

