

# 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 := "neither";  
      }  
    } else {  
      s := "neither";  
    }  
  }  
}
```

```

        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)