

Experiment ‘p4_2_b_count_even_odd’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY

Bad responses: 0

Responses containing assume : 0

Resolution attempts: 8

Hard fails (resolution): 0

Soft fails (resolution): 1

Verification attempts: 7

Problem Specification

Problem name: p4_2_b_count_even_odd

Natural language statement: Write a method that takes a sequence of integers and returns the number of even and odd inputs.

Method signature: p4_2_b_count_even_odd(inputs: seq<int>) returns (even_count: int, odd_count: int)

Ensures

- even_count == number_even(inputs[...])
- odd_count == number_odd(inputs[...])

Functional Code Given

```
function number_odd(s:seq<int>): int
{ if s == [] then 0
  else if s[0] % 2 == 1 then 1 + number_odd(s[1..])
  else number_odd(s[1..])
}

function number_even (s:seq<int>): int
{ if s == [] then 0
  else if s[0] % 2 == 0 then 1 + number_even(s[1..])
  else number_even(s[1..])
}
```

GenAI interactions

Below you will find all interactions between the ‘user’ (program) and the ‘assistant’ (OpenAI).

Program → GenAI

You are given the following task to perform in Dafny:

Write a method that takes a sequence of integers and returns the number of even and odd
→ inputs.

The signature should be:

```
method p4_2_b_count_even_odd(inputs: seq<int>) returns (even_count: int, odd_count: int)
```

The method should respect the following contract:

```
ensures even_count == number_even(inputs[..]), ensures odd_count == number_odd(inputs  
→ [..])
```

The contract uses the following dafny code:

```
function number_odd(s:seq<int>): int  
{ if s == [] then 0  
  else if s[0] % 2 == 1 then 1 + number_odd(s[1..])  
  else number_odd(s[1..])  
}  
  
function number_even (s:seq<int>): int  
{ if s == [] then 0  
  else if s[0] % 2 == 0 then 1 + number_even(s[1..])  
  else number_even(s[1..])  
}
```

These function(s) must not be used in your implementation of the method.

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

GenAI → Program

System fingerprint:

ID: chatmpl-Co4aVu7EgVtcsZdUDVWuCi5gWcfgb

Model: gpt-5.1-chat-latest

Created at: 1766050071

Finish reason: stop

Usage: 568 tokens in, and 340 tokens out

```
{  
  var i := 0;  
  even_count := 0;
```

```

odd_count := 0;
while i < |inputs|
    invariant 0 <= i <= |inputs|
    invariant even_count == number_even(inputs[..i])
    invariant odd_count == number_odd(inputs[..i])
{
    if inputs[i] % 2 == 0 {
        even_count := even_count + 1;
    } else {
        odd_count := odd_count + 1;
    }
    i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
    ↪ proved on this return path
|
16 | {
| ^
|   ensures even_count == number_even(inputs[..])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(14,20): Related location: this is the
    ↪ postcondition that could not be proved
|
14 |   ensures even_count == number_even(inputs[..])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
    ↪ proved on this return path
|
16 | {
| ^
|   ensures odd_count == number_odd(inputs[..])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(15,19): Related location: this is the
    ↪ postcondition that could not be proved
|
15 |   ensures odd_count == number_odd(inputs[..])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(22,25): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
22 | invariant even_count == number_even(inputs[..i])
| ^
|   ^

```

```

problem_data/p4_2_b_count_even_odd/program.dfy(23,24): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
23 | invariant odd_count == number_odd(inputs[..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: chatmpl-Co4abHvQAEHwYbSveIPJOBpkCMPg5
Model: gpt-5.1-chat-latest
Created at: 1766050077
Finish reason: stop
Usage: 1233 tokens in, and 220 tokens out

```
{
    var i := 0;
    even_count := 0;
    odd_count := 0;
    while i < |inputs|
        invariant 0 <= i <= |inputs|
        invariant even_count == number_even(inputs[..i])
        invariant odd_count == number_odd(inputs[..i])
        decreases |inputs| - i
    {
        if inputs[i] % 2 == 0 {
            even_count := even_count + 1;
        } else {
            odd_count := odd_count + 1;
        }
        i := i + 1;
    }
}
```

Program → GenAI

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

```

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
    ↪ proved on this return path
|
16 | {
```

```

problem_data/p4_2_b_count_even_odd/program.dfy(14,20): Related location: this is the
    ↪ postcondition that could not be proved
|
14 |   ensures even_count == number_even(inputs[...])
|   ^
|
problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
    ↪ proved on this return path
|
16 | {
| ^
|
problem_data/p4_2_b_count_even_odd/program.dfy(15,19): Related location: this is the
    ↪ postcondition that could not be proved
|
15 |   ensures odd_count == number_odd(inputs[...])
|   ^
|
problem_data/p4_2_b_count_even_odd/program.dfy(22,25): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
22 | invariant even_count == number_even(inputs[..i])
|   ^
|
problem_data/p4_2_b_count_even_odd/program.dfy(23,24): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
23 | invariant odd_count == number_odd(inputs[..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: chatmpl-Co4afsSJsMuobSr5iynqoLPQb9Dw4

Model: gpt-5.1-chat-latest

Created at: 1766050081

Finish reason: stop

Usage: 1781 tokens in, and 164 tokens out

```
{
  var i := 0;
```

```

even_count := 0;
odd_count := 0;
while i < |inputs|
    invariant 0 <= i <= |inputs|
    invariant inputs == old(inputs)
    invariant even_count == number_even(inputs[..i])
    invariant odd_count == number_odd(inputs[..i])
    decreases |inputs| - i
{
    if inputs[i] % 2 == 0 {
        even_count := even_count + 1;
    } else {
        odd_count := odd_count + 1;
    }
    i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p4_2_b_count_even_odd/program.dfy(22,24): Warning: Argument to 'old' does
    ↪ not dereference the mutable heap, so this use of 'old' has no effect
    |
22 | invariant inputs == old(inputs)
    | ^^^

```

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

Correct the error by altering only the method body. Produce and show only the Dafny body,
 ↪ including the curly braces that surround it. Do not show the signature nor
 ↪ contract. You must not use 'assume' anywhere in your code.

GenAI → Program

System fingerprint:

ID: chatcmpl-Co4ajEvYAhrU7f0DckGmVc55F5qf

Model: gpt-5.1-chat-latest

Created at: 1766050085

Finish reason: stop

Usage: 1831 tokens in, and 156 tokens out

```
{
    var i := 0;
    even_count := 0;
    odd_count := 0;
    while i < |inputs|
        invariant 0 <= i <= |inputs|
        invariant even_count == number_even(inputs[..i])
        invariant odd_count == number_odd(inputs[..i])
}
```

```

decreases |inputs| - i
{
  if inputs[i] % 2 == 0 {
    even_count := even_count + 1;
  } else {
    odd_count := odd_count + 1;
  }
  i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
  ↪ proved on this return path
  |
16 | {
  | ^
  |

problem_data/p4_2_b_count_even_odd/program.dfy(14,20): Related location: this is the
  ↪ postcondition that could not be proved
  |
14 | ensures even_count == number_even(inputs[...])
  | ^^

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
  ↪ proved on this return path
  |
16 | {
  | ^
  |

problem_data/p4_2_b_count_even_odd/program.dfy(15,19): Related location: this is the
  ↪ postcondition that could not be proved
  |
15 | ensures odd_count == number_odd(inputs[...])
  | ^^

problem_data/p4_2_b_count_even_odd/program.dfy(22,25): Error: this invariant could not be
  ↪ proved to be maintained by the loop
  Related message: loop invariant violation
  |
22 | invariant even_count == number_even(inputs[..i])
  | ^^

problem_data/p4_2_b_count_even_odd/program.dfy(23,24): Error: this invariant could not be
  ↪ proved to be maintained by the loop
  Related message: loop invariant violation
  |
23 | invariant odd_count == number_odd(inputs[..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-Co4anBgVsr0p57jCuLCFyznOhr0tl

Model: gpt-5.1-chat-latest

Created at: 1766050089

Finish reason: stop

Usage: 2632 tokens in, and 220 tokens out

```
{  
    var i := 0;  
    even_count := 0;  
    odd_count := 0;  
    while i < |inputs|  
        invariant 0 <= i <= |inputs|  
        invariant even_count == number_even(inputs[..i])  
        invariant odd_count == number_odd(inputs[..i])  
        decreases |inputs| - i  
    {  
        if inputs[i] % 2 == 0 {  
            even_count := even_count + 1;  
        } else {  
            odd_count := odd_count + 1;  
        }  
        i := i + 1;  
    }  
}
```

Program → GenAI

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

```
problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be  
↳ proved on this return path  
|  
16 | {  
| ^  
  
problem_data/p4_2_b_count_even_odd/program.dfy(14,20): Related location: this is the  
↳ postcondition that could not be proved  
|  
14 | ensures even_count == number_even(inputs[...])
```

```

| ^^

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
    ↪ proved on this return path
|
16 | {
| ^

problem_data/p4_2_b_count_even_odd/program.dfy(15,19): Related location: this is the
    ↪ postcondition that could not be proved
|
15 | ensures odd_count == number_odd(inputs[..])
| ^^

problem_data/p4_2_b_count_even_odd/program.dfy(22,25): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
22 | invariant even_count == number_even(inputs[..i])
| ^^

problem_data/p4_2_b_count_even_odd/program.dfy(23,24): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
23 | invariant odd_count == number_odd(inputs[..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: chatmpl-Co4asSOTKqWRay0PNQ6BKeaRdTQSL

Model: gpt-5.1-chat-latest

Created at: 1766050094

Finish reason: stop

Usage: 3180 tokens in, and 422 tokens out

```
{
var i := 0;
even_count := 0;
odd_count := 0;
while i < |inputs|
    invariant 0 <= i <= |inputs|
    invariant even_count == number_even(inputs[..i])
```

```

invariant odd_count == number_odd(inputs[..i])
decreases |inputs| - i
{
  if inputs[i] % 2 == 0 {
    even_count := even_count + 1;
  } else if inputs[i] % 2 == 1 {
    odd_count := odd_count + 1;
  }
  i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
  ↪ proved on this return path
  |
16 | {
  | ^
  |

problem_data/p4_2_b_count_even_odd/program.dfy(14,20): Related location: this is the
  ↪ postcondition that could not be proved
  |
14 | ensures even_count == number_even(inputs[..])
  | ^~

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
  ↪ proved on this return path
  |
16 | {
  | ^
  |

problem_data/p4_2_b_count_even_odd/program.dfy(15,19): Related location: this is the
  ↪ postcondition that could not be proved
  |
15 | ensures odd_count == number_odd(inputs[..])
  | ^~

problem_data/p4_2_b_count_even_odd/program.dfy(22,25): Error: this invariant could not be
  ↪ proved to be maintained by the loop
Related message: loop invariant violation
  |
22 | invariant even_count == number_even(inputs[..i])
  | ^~

problem_data/p4_2_b_count_even_odd/program.dfy(23,24): Error: this invariant could not be
  ↪ proved to be maintained by the loop
Related message: loop invariant violation
  |

```

```
23 | invariant odd_count == number_odd(inputs[..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-Co4azyXeRhxngSb6oykW6rGSJE1F

Model: gpt-5.1-chat-latest

Created at: 1766050101

Finish reason: stop

Usage: 3738 tokens in, and 230 tokens out

```
{
    var i := 0;
    even_count := 0;
    odd_count := 0;
    while i < |inputs|
        invariant 0 <= i <= |inputs|
        invariant even_count + odd_count == i
        invariant even_count == number_even(inputs[..i])
        invariant odd_count == number_odd(inputs[..i])
        decreases |inputs| - i
    {
        if inputs[i] % 2 == 0 {
            even_count := even_count + 1;
        } else {
            odd_count := odd_count + 1;
        }
        i := i + 1;
    }
}
```

Program → GenAI

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

```
problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
    ↳ proved on this return path
    |
16 | {
    | ^

problem_data/p4_2_b_count_even_odd/program.dfy(14,20): Related location: this is the
    ↳ postcondition that could not be proved
```

```

14 |   ensures even_count == number_even(inputs[..])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(16,0): Error: a postcondition could not be
  ↪ proved on this return path
|
16 | {
| ^

problem_data/p4_2_b_count_even_odd/program.dfy(15,19): Related location: this is the
  ↪ postcondition that could not be proved
|
15 |   ensures odd_count == number_odd(inputs[..])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(23,25): Error: this invariant could not be
  ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
23 | invariant even_count == number_even(inputs[..i])
|   ^~

problem_data/p4_2_b_count_even_odd/program.dfy(24,24): Error: this invariant could not be
  ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
24 | invariant odd_count == number_odd(inputs[..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: chatmpl-Co4b4C1Ou6QGdit5Cr8HeOYkeerPr

Model: gpt-5.1-chat-latest

Created at: 1766050106

Finish reason: stop

Usage: 4296 tokens in, and 176 tokens out

```
{
  var i := 0;
  even_count := 0;
  odd_count := 0;
  while i < |inputs|
```

```

invariant 0 <= i <= |inputs|
invariant even_count == number_even(inputs[..i])
invariant odd_count == number_odd(inputs[..i])
decreases |inputs| - i
{
    // First advance i so the slice inputs[..i] includes the next element
    i := i + 1;

    if inputs[i-1] % 2 == 0 {
        even_count := even_count + 1;
    } else {
        odd_count := odd_count + 1;
    }
}
}

```

Final Program

```

function number_odd(s:seq<int>): int
{ if s == [] then 0
  else if s[0] % 2 == 1 then 1 + number_odd(s[1..])
  else number_odd(s[1..])
}

function number_even (s:seq<int>): int
{ if s == [] then 0
  else if s[0] % 2 == 0 then 1 + number_even(s[1..])
  else number_even(s[1..])
}

method p4_2_b_count_even_odd(inputs: seq<int>) returns (even_count: int, odd_count: int)
    ensures even_count == number_even(inputs[..])
    ensures odd_count == number_odd(inputs[..])
{
    var i := 0;
    even_count := 0;
    odd_count := 0;
    while i < |inputs|
        invariant 0 <= i <= |inputs|
        invariant even_count == number_even(inputs[..i])
        invariant odd_count == number_odd(inputs[..i])
        decreases |inputs| - i
    {
        // First advance i so the slice inputs[..i] includes the next element
        i := i + 1;

        if inputs[i-1] % 2 == 0 {
            even_count := even_count + 1;
        } else {
            odd_count := odd_count + 1;
        }
    }
}

```

```
    }  
}  
}
```

Total Token Usage

Input tokens: 19259

Output tokens: 1928

Reasoning tokens: 640

Sum of ‘total tokens’: 21187

Experiment Timings

Iteration #8 started at 1766050108055, ended at 1766050112755, lasting 4700ms (4.70 seconds)

Overall Experiment started at 1766050073740, ended at 1766050112755, lasting 39015ms (39.02 seconds)

Iteration #4 started at 1766050087002, ended at 1766050090765, lasting 3763ms (3.76 seconds)

Iteration #5 started at 1766050090765, ended at 1766050095776, lasting 5011ms (5.01 seconds)

Iteration #6 started at 1766050095776, ended at 1766050102910, lasting 7134ms (7.13 seconds)

Iteration #7 started at 1766050102910, ended at 1766050108055, lasting 5145ms (5.15 seconds)

Iteration #1 started at 1766050073741, ended at 1766050078804, lasting 5063ms (5.06 seconds)

Iteration #2 started at 1766050078804, ended at 1766050083770, lasting 4966ms (4.97 seconds)

Iteration #3 started at 1766050083770, ended at 1766050087002, lasting 3232ms (3.23 seconds)