

# Exercises 4: Dynamic Logic



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

The solutions to the exercises will be discussed on Monday, 22nd June.

---

## Problem 1

---

What does the following method do, assuming that  $x$  is a non-negative integer?

```
public int x;  
  
public int method1() {  
    int y = x;  
    int z = 0;  
    while (y > 0) {  
        z = z + x;  
        y = y - 1;  
    }  
    return z;  
}
```

- a) Write JML specifications for method1 and invariants for the class. (The Java code is given in file loops/Loop1.java)
- b) Prove partial correctness for method1 by providing a loop invariant.
- c) Is total correctness possible to prove? Justify!

---

## Problem 2

---

What does the following method do, assuming that  $x$  is a non-negative integer and  $y$  is a strictly positive integer?

```
public int x;  
public int y;  
  
public int method2() {  
    int x1 = x, q = 0;  
  
    while (x1 >= y) {  
        x1 = x1 - y;  
        q = q + 1;  
    }  
    return q;  
}
```

- a) Write JML specifications for method2 and invariants for the class. (The Java code is given in file loops/Loop2.java)
- b) Prove partial correctness for method2 by providing a loop invariant.
- c) Is total correctness possible to prove? Justify!

---

### Problem 3

---

#### Problem 3.1 Array Reversal

---

Open file `loops/ReverseArray.java` in an editor. The class contains the static method `reverse` which gets an array `p_a` as argument and returns a new array of the same length but with the elements of `p_a` in reverse order.

- Specify the method as precisely as possible in JML.
- Specify the loop invariant as strong as necessary to prove the method contract.
- Verify that the method is correct w.r.t. its contract.

---

#### Problem 3.2 In-Place reversal

---

Open file `loops/ReverseArray.java` in an editor. The class contains a method stub `reverseInPlace()`. Calling the method should perform an in-place reversal of the array contained in field `a`.

- Specify the method behavior as described as precisely as possible in JML.
- Implement the method and add, if necessary, loop invariant specifications.
- Verify that the method is correct w.r.t. its contract.

---

#### Problem 3.3 Binary Search

---

Open file `loops/ReverseArray.java` in an editor.

The method `search(int e, int[] sortedArray)` checks whether the provided element is contained in array `sortedArray` using the recursive method `searchHelper`. The method requires that the given array is sorted in ascending order. Specify method `|searchHelper`. and prove the total correctness of both methods.