

Nennen Sie eine möglichst genaue Nachbedingung (Beispiel 1). Bei diesem Code handelt es sich um Java-Code. Beachten Sie das bei Ihrer Antwort!

```
int x;

// Precondition: x = 5
x = x + 1;
// Postcondition?
```

Nennen Sie eine möglichst genaue Nachbedingung (Beispiel 2). Bei diesem Code handelt es sich um Java-Code. Beachten Sie das bei Ihrer Antwort!

```
int x;

// Precondition: x >= 0
x = x + 1;
// Postcondition?
```

Stack-Problem. Gegeben sei ein Stack und die Zahlen 1 bis 4. Sie dürfen auf dem Stack **push**- und **pop**-Operationen ausführen, wobei Sie immer nur die kleinste bislang noch nicht gepushte Zahl pushen dürfen. (Ist also 1 bereits gepusht, dürfen Sie entweder poppen oder 2 pushen, aber nicht 1, 3 oder 4.) Gibt es eine Folge von **push**- und **pop**-Operationen, bei der Sie die Zahlen 2, 4, 3 und 1 in dieser Reihenfolge poppen? Können Sie die Zahlen 4, 2, 3 und 1 in dieser Reihenfolge poppen? Begründen Sie Ihre Antwort.

Ringbuffer-Problem. In einem Ringbuffer der Grösse $n = 10$ sind `outIdx = 1` und `inIdx = 2`. Wieviele Elemente sind in diesem Ringbuffer gespeichert? Beantworten Sie diese Frage auch für die entsprechenden Werte in der unten stehenden Tabelle.

outIdx	inIdx	Anzahl?
1	2	
1	3	
1	0	
1	1	

Welche Laufzeit hat folgender Code (Beispiel 3)?

```
public int computeSum(int n) {
    int sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += i;
    }
    return sum;
}
```

Welche Laufzeit hat folgender Code (Beispiel 4)?

```
public int computeSum(int m, int n) {  
    int sum = 0;  
    for (int i = 1; i <= n; i++) {  
        for (int j = 1; j <= m; j++) {  
            sum += i*j;  
        }  
    }  
    return sum;  
}
```

Welche Laufzeit hat folgender Code (Beispiel 5)?

```
public int computeSum(int n) {  
    int sum = 0;  
    for (int i = 1; i <= n; i++) {  
        for (int j = 1; j <= i; j++) {  
            sum += i*j;  
        }  
    }  
    return sum;  
}
```

Hat folgender Code Laufzeit $O(n^3)$ (Beispiel 6)?

```
public int computeSum(int n) {  
    int sum = 0;  
    for (int i = 1; i <= n; i++) {  
        for (int j = 1; j <= n; j++) {  
            sum += i*j;  
        }  
    }  
    return sum;  
}
```

Welche Laufzeit hat folgender Code (Beispiel 7)?

```
public void countOccurrences(int[] F) {  
    // Assume that F is sorted.  
    int count = 0;  
    for (int i = 0; i < F.length; i++) {  
        if (binarySearch(F, i)) {  
            count++;  
        }  
    }  
    return count;  
}
```

Pseudo-Code zur Sentinel-Frage

```
public ListItem searchUnsorted(IntegerList l, int x) {
    ListItem p = l.getFirst();

    while (p != null && p.getValue() != x) {
        p = p.getNext();
    }

    return p;
}

public ListItem searchUnsortedWithSentinel(IntegerList l, int x) {
    ListItem sentinel = l.appendAtEnd(x); // Insert x as sentinel
    ListItem p = l.getFirst();

    while (p.getValue() != x) {
        p = p.getNext();
    }

    ListItem ret = p == sentinel ? null : p;

    l.remove(sentinel); // remove sentinel again
    return ret;
}
```

Code-Beispiel 8.

```
public int f(int n) {
    if (n == 0) {
        return 0;
    } else if (n == 1) {
        return 1;
    } else {
        return 2*f(n-1) + 3*f(n-2);
    }
}
```

Beginnt mit genau 2!

$O(2^n)$

1 0 2 1 3 2

Code-Beispiel 9.

```
static int binary(int[] a, int s) {
    int l = -1;
    int r = a.length;

    // inv es l == -1 es r == a.length
    while (l != r && a[l] != s) {
        int m = (l + r) / 2;

        if (a[m] < s) {
            l = m;
        } else if (a[m] > s) {
            r = m;
        }
    }

    // inv es (l == r || a[l] == s)
    return a[l] == s ? l : -1;
}
```

Code-Beispiel 10.

```
/** Computes the union of two sets.
 *
 * @param a one of the two sets in the union
 * @param b one of the two sets in the union
 * @returns the union of {@code a} and {@code b}.
 */
static Set<Integer> union(Set<Integer> a, b) {
    Set<Integer> ret = new HashSet<>();

    ret.addAll(a);

    for (Integer x : b) {
        if (!ret.contains(x)) {
            ret.add(x);
        }
    }

    return ret;
}
```