

Aufgabenblatt 1: Prozedurales Java

Aufgabe 1

a)

```
[1, 2, 17, 4, -1, -2, -3, -4]
```

b)

Teil A

- Fehler in Zeile #9: `i <= a.length`
- Richtig:

```
for (int i = 0; i < a.length; i++)
```

Teil B

- ist korrekt

Teil C

- Fehler in Zeile #23: `m[i]`
- Richtig:

```
m[i - 1] = val / 3;
```

Teil D

- Fehler in Zeile #27: `max = 0`
- Richtig:

```
int max = Integer.MIN_VALUE;
```

Teil E

- Fehler in Zeile #36: `a[i] != 0`
- Richtig:

```
while (i < a.length && a[i] == 0) i++;
```

c)

```
int a1[] = new int[a.length % 2 == 0; ? a.length / 2 : a.length / 2 + 1],  
a2[] = new int[a.length / 2];  
for (int i = 0; i < a.length; i++) {  
    if (i % 2 == 0)  
        a1[i / 2] = a[i];  
    else  
        a2[i / 2] = a[i];  
}
```

d)

```
int oddCount = 0;  
int evenCount = 0;  
for (int i = 0; i < a.length; i++) {  
    if (a[i] % 2 == 0)  
        evenCount++;  
    else  
        oddCount++;  
}  
  
int b1[] = new int[evenCount], b2[] = new int[oddCount];  
for (int i = 0; i < a.length; i++) {  
    if (a[i] % 2 == 0)  
        b1[i / 2] = a[i];  
    else  
        b2[i / 2] = a[i];  
}
```

Aufgabe 2

```
public class Fibonacci {  
    public static void main(String[] args) {  
        int n = Integer.parseInt(args[0]);  
        if (n == 0)  
            System.out.println(0);  
  
        if (n <= 0)  
            System.exit(1);  
  
        int previous = 0;  
        int fibonacci = 1;  
  
        for (int i = 0; i < n; i++) {  
            int newPrevious = fibonacci;  
            fibonacci += previous;  
            previous = newPrevious;  
        }  
  
        System.out.println(fibonacci);  
    }  
}
```

Aufgabe 3

```
public static int prost(int n) {  
    if (n == 1) return 0;  
    if (n == 2) return 1;  
    return n - 1 + prost(n - 1);  
}
```

In der Aufgabenstellung wurde beschrieben, dass sich die Anzahl der Anstöße mithilfe der Summe der ersten n Zahlen berechnen lässt. Allerdings gibt es bei nur **einer** einzelnen Person keine Anstöße.
Deswegen `if (n == 1) return 0.`