

# Lösningsförslag övning 3.

## Uppgift 1

a)

```
//before: a != null && b != null
//before: a[i].length == a[0].length for all i
//before: a[i].length == b[i].length for all i
public int[][] matrixAddition(int[][] a, int[][] b) {
    int[][] c = new int[a.length][a[0].length];
    for(int i = 0; i < a.length; i++) {
        for(int j = 0; j < a[0].length; j++) {
            c[i][j] = a[i][j] + b[i][j];
        }
    }
    return c;
} //matrixAddition
```

b)

```
//before: matrix != null && matrix[i].length == matrix[0].length for all i
public int[][] transpose(int[][] matrix) {
    int[][] result = new int[matrix[0].length][matrix.length];
    for(int i = 0; i < matrix[0].length; i++) {
        for(int j = 0; j < matrix.length; j++) {
            result[i][j] = matrix[j][i];
        }
    }
} //transpose
```

c)

```
//before: a != null && b != null
//before: a[i].length == a[0].length for all i
//before: b[i].length == b[0].length for all i
//before: a[i].length == b.length for all i
public int[][] multiply(int[][] a, int[][] b) {
    int[][] result = new int[a.length][b[0].length];
    for(int i = 0; i < a.length; i++) {
        for(int j = 0; j < b[0].length; j++) {
            int sum = 0;
            for(int k = 0; k < b.length; k++) {
                sum += a[i][k] * b[k][j];
            }
            result[i][j] = sum;
        }
    }
    return result;
} //multiply
```

## Uppgift 2

a)

```
//before: none
public int countElements(int[][] array) {
    int result = 0;
    for(int i = 0; i < array.length; i++) {
        result += array[i].length;
    }
    return result;
} //countElements
```

b)

```
//before: none
public int sumElements(int[][] array) {
    int sum = 0;
    for(int i = 0; i < array.length; i++) {
        for(int j = 0; j < array[i].length; j++) {
            sum += array[i][j];
        }
    }
} //sumElements
```

## Uppgift 3

```
//before: a != null && b != null
public static ArrayList<Integer> append(ArrayList<Integer> a, ArrayList<Integer> b) {
    ArrayList<Integer> res = new ArrayList<Integer>();
    for (int pos = 0; pos < a.size(); pos++)
        res.add(a.get(pos));
    for (int pos = 0; pos < b.size(); pos++)
        res.add(b.get(pos));
    return res;
} //append
```

### Alternativ lösning:

```
//before: a != null && b != null
public static ArrayList<Integer> append(ArrayList<Integer> a, ArrayList<Integer> b) {
    ArrayList<Integer> res = new ArrayList<Integer>();
    for (Integer elm : a)
        res.add(elm);
    for (Integer elm : b)
        res.add(elm);
    return res;
} //append
```

### Alternativ lösning:

```
//before: a != null && b != null
public static ArrayList<Integer> append(ArrayList<Integer> a, ArrayList<Integer> b) {
    ArrayList<Integer> res = new ArrayList<Integer>(a);
    res.addAll(b);
    return res;
} //append
```

#### Uppgift 4

```
//before: a != null && b != null
public static ArrayList<Integer> merge(ArrayList<Integer> a, ArrayList<Integer> b) {
    ArrayList<Integer> res = new ArrayList<Integer>();
    int pos = 0;
    while (pos < a.size() && pos < b.size()) {
        res.add(a.get(pos));
        res.add(b.get(pos));
        pos++;
    }
    for (int i = pos; i < a.size(); i++)
        res.add(a.get(i));
    for (int i = pos; i < b.size(); i++)
        res.add(b.get(i));
    return res;
} //merge
```

#### Uppgift 5

```
//before: a != null && b != null
public static ArrayList<Integer> mergeSorted(ArrayList<Integer> a, ArrayList<Integer> b) {
    ArrayList<Integer> res = new ArrayList<Integer>();
    int pos1 = 0;
    int pos2 = 0;
    while (pos1 < a.size() && pos2 < b.size()) {
        if (a.get(pos1) < b.get(pos2)) {
            res.add(a.get(pos1));
            pos1++;
        }
        else {
            res.add(b.get(pos2));
            pos2++;
        }
    }
    for (int i = pos1; i < a.size(); i++)
        res.add(a.get(i));
    for (int i = pos2; i < b.size(); i++)
        res.add(b.get(i));
    return res;
} //mergeSorted
```

## Uppgift 6

a)

```
//before: people != null
public static ArrayList<String> bornThisYear(ArrayList<Person> people, int year) {
    ArrayList<String> res = new ArrayList<String>();
    for (Person p : people) {
        if (p.getYear() == year)
            res.add(p.getName());
    }
    return res;
} //bornThisYear
```

b)

```
//before: people != null
public static void removeNames(ArrayList<Person> people, ArrayList<String> names) {
    int pos = 0;
    while (pos < people.size()) {
        if (isInList(names, people.get(pos).getName()))
            people.remove(pos);
        else
            pos++;
    }
} //removeNames

//before: names != null
private static boolean isInList(ArrayList<String> names, String name) {
    for (String elem: names) {
        if (elem.equals(name))
            return true;
    }
    return false;
} //isInList
```

Diskutera/förklara varför inte följande ”lösningsförslag” fungerar?

```
//before: people != null
public static void removeNames(ArrayList<Person> people, ArrayList<String> names) {
    for (int pos = 0; pos < people.size(); pos++) {
        if (isInList(names, people.get(pos).getName())) {
            people.remove(pos);
        }
    }
} //removeNames
```