

# Övning 4

II

a)

```
public static int min(int[] v) {
    int min = Math.MAX_VALUE;
    for (int i = 0; i < v.length; i++) {
        if (v[i] < min)
            v[i] = min;
    }
    return min;
}
```

b)

```
public static boolean isSorted(int[] v) {
    boolean isSorted = true;
    for (int i = 1; i < v.length; i++) {
        if (v[i] < v[i-1]) {
            isSorted = false;
        }
    }
    return isSorted;
}
```

1

```
c) public static boolean equal(int[] v1, int[] v2)
{
    if (v1.length != v2.length) {
        return false;
    }
    for (int i = 0; i < v1.length; i++) {
        if (v1[i] != v2[i]) {
            return false;
        }
    }
    return true;
}
```

```

2 public class Matrix {
    private double[][] a;

    public Matrix(int n) {
        a = new double[n][n];
    }

    public double get(int i, int k) {
        return a[i][k];
    }

    public void add(Matrix m) {
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length; j++) {
                a[i][j] += m[i][j];
            }
        }
    }

    public double trace() {
        double sum = 0;
        for (int i = 0; i < a.length; i++) {
            sum += a[i][i];
        }
        return sum;
    }
}

```

3

a) 33 i blir 33 i 32:a loopen.

b) 100 Hundra loopar...

c) 1 Den itererar bara en gång innan  
 $a[i] = 0$  då  $i = 1$

---

5 `int[] nbrDigits(int start, int stop, int interval){`

`int nbr;`  
`int[] digits = new int[10];`

`for (int i = start; i <= stop; i += interval){`

`nbr = i;`

`while (nbr != 0){`

`digits[nbr % 10]++;`

`nbr = nbr / 10;`

`}`

`}`

`return digits;`

`}`

6

```
public class Map {  
    private Town[] towns;
```

```
    public Map(string mapFile) {
```

DONE

```
}
```

```
    public void printTour(string startTown) {
```

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

```
            towns[i].setVisited(false);
```

```
        }
```

```
        int i=0;
```

```
        int start=0;
```

```
        while (towns[i].getName().equals(startTown)) {
```

```
            start++;
```

```
        }
```

```
        towns[start].setVisited(true)
```

```
        System.out.println(towns[start].getName);
```

```
        Town current = towns[start];
```

```
        current = current.getNearest(current);
```

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

```
            current = current getNearest(current);
```

```
            System.out.println(current.getName);
```

```
            current.setVisited(true);
```

```
        }
```

Letar upp närmaste besökta!

```
private Town getNearest(Town t) {
```

```
    double dist = Math.MAX_VALUE
```

```
    Town nearest = null;
```

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

```
        if (t.getDistanceTo(towns[i]) < dist
```

```
            && towns[i] != t && !towns[i].isVisited()) {
```

```
            dist = t.getDistanceTo(towns[i])
```

```
            nearest = towns[i]
```

```
        }
```

```
    }
```

```
    return nearest;
```

```
}
```

```
}
```