Prof. Ralf Schenkel   Wintersemester 21/22
Tobias Zeimetz
Trier University

Exercises for the Class
Elements of Computer Science: Programming
Live Assignment 02
Submission of solutions until 6:00 p.m.
at `moodle.uni-trier.de`

- Submission that can't be compiled are rated with **0** points!

- Please comment your solutions, otherwise you can lose points!

Exercise 1 (Evaluation: Text)   (10 Points)

Write a program that reads three `int` numbers n, m and k and outputs a list of the numbers between n and m (both inclusive). However, any number that is divisible by k without a remainder should be replaced by the text `beep`.

For the case $n < m$, the list should be sorted in ascending order, as in the following example:

```
Input:  4 10 3
Output: 4 5 beep 7 8 beep 10
```

For the case $n > m$, the list should be sorted in descending order, as in the following example:

```
Input:  10 3 2
Output: beep 9 beep 7 beep 5 beep 3
```


Exercise 2 (Evaluation: Text)   (10 Points)

Write a program that can read 5 numbers via the scanner. These should then be used to fill an array `array` of size 5.

Then you have to count how many even and odd numbers are contained in this field and display them on the console as follows.

```
Input: 5 2 3 4 8
Even: 3
Odd: 2
```

Exercise 3 (Evaluation: Numbers)   (10 Points)

In the given initial part of the program first an integer `number` (with value > 0) is read in and then an array `int[] field` of size `number` is created. This array is then filled with random numbers by initializing a random number generator with a second value, given by the user. The numbers in the array can be positive or negative. Up to this point the program is given and may not be changed.

   You should now find out (a) how many of the numbers in the array are less than 0 and (b) how many are greater than 0. As an example, if you enter `3 3` as to the program, your array contains the following numbers: `4 -1 6`

   Therefore your output has to look as follows:

```
Size: 3
Seed: 3
Negatives: 1
Positives: 2
```

Exercise 4 (Evaluation: Numbers/Text)   (10 Points)

Write a program that takes several **int** values as input. The first input (called `size`) is used to determine the size of an array `array`. Subsequently, the user has to pass as many numbers to the program as the array is sized.

   After the array is filled with numbers given by a user, the program has to find the greatest index in the array for which yields `array[index] == index`.

```
Input: 5 0 1 2 3 4
Output: 4

Input: 4 1 1 2 6
Output: 2
```

   If your array does not contain values so that the constraint `array[index] == index` yields, print your output as follows:

```
Input: 5 1 2 3 4 5
Output: No Index Found
```

Exercise 5 (Evaluation: Numbers)   (10 Points)

Write a Java program that reads a positive number `x` from a dialog with the user and returns the number inverted digit by digit. For example, your program should just return 54321 for the input 12345. Think about how the modulo-operator (`%`) and the divisor (`/`) (e.g. `x%10` and `x/10`) can help you!

   You may assume that the read number `x` has at least 2 digits.