Prof. Ralf Schenkel
Tobias Zeimetz
Trier University

Exercises for the Class
Elements of Computer Science: Programming
Live Assignment 03

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: Numbers/Text) (10 Points)

Your program should first read in a `int` number a (where you can assume that $a > 0$ is).
Then it should preset a `double` variable x with the value 2 and then with a `for` loop ten
times (a) first output the current value of x and (b) then redefine x each time via the formula
$x = (x + a/x)/2$ (similar to the Lunar Land Game without protocol). Exactly 10 `double`
numbers are printed.

Example:

```
Input: 16
Output: 2.0 5.0 4.1 4.001219512195122 4.0000001858445895
4.000000000000004 4.0 4.0 4.0 4.0
```

Note: This method is the so-called "Heron" method for calculating the square root.

Exercise 2 (Evaluation: Numbers) (10 Points)

The given progam reads in two arrays. In this task the arrays should be interpreted as vectors.
Your task is to determine the scalar product of both vectors. Use the following formula:
$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \text{ and } \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \text{ lead to } \vec{a} \circ \vec{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$$
An example of the application of the scalar product is:
$$\vec{a} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} \text{ and } \vec{b} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} \text{ lead to } \vec{a} \circ \vec{b} = 2 \times 10 + 3 \times 20 + 4 \times 30 = 200$$

You can assume, that both read in arrays have the same size.

Exercise 3 (Evaluation: Text)   (15 Points)

Write a program that reads a **int** number as input and generates a pattern (from $n \times n$ characters) of the following type as output (similar to the example `K3B14E_Flag`):

Input: **6** **XXXXXX** **XXXXXo** **XXXXoo** **XXXooo** **XXoooo** **Xooooo**	Input: **7** **XXXXXXX** **XXXXXXo** **XXXXXoo** **XXXXooo** **XXXoooo** **XXooooo** **Xoooooo**

Also make sure that no line ends are missing and that you do not create unnecessary blank lines, otherwise the evaluation will not work. The letters used in the pattern are the capital X and the lower o.

Exercise 4 (No Evaluation)   (15 Points)

Use your program to find the smallest positive int number n that is so large that the value $n \cdot n$ can no longer be represented as a int value (called overflow).

Hint: Which values would $(n \cdot n)/n$ and $(n \cdot n)\%n$ have to have, if calculated correctly? An input is not needed, the output should have the following form:

First overflow during squaring occurs at **50000**

Watch out: The value **50000** is not the correct result. *During the exercise* there is no correct value for the evaluation given, i.e. you cannot test your program for correctness by the evaluation.