Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecturer: Prof. B. Gärtner

Informatik für Mathematiker und Physiker HS16

Exercise Sheet 4

Submission deadline: 15:15 - Tuesday 18th October, 2016
Course URL: http://lec.inf.ethz.ch/ifmp/2016/

Assignment 1 – Determining Loop Outputs (4 points)

Determine the outputs of the following loops!

a) What does the following loop output?

```
unsigned int x = 5;
while (x > 0) {
    std::cout << x << " ";
    --x;
}
```

b) What does the following loop output? [based on Exam Summer 2016]

```
unsigned int x = 4;
do {
    std::cout << x << " ";
    x = (2 * x + 3) % 13;
} while (x != 0);
```

c) What does the following loop output?

```
for (int i = 0; i <= 10; i += 2) {
    if (i % 4 == 0)
        continue;
    std::cout << i << " ";
}
```

d) What does the following loop output? [based on Exam Summer 2012]

```
for (int i=1; i<=3; ++i)
    for (int j=1; j<=3; ++j)
        std::cout << "(" << i << "," << j << ")";
```

Assignment 2 – Writing C++ Loops (8 points)

In this exercise you are given 4 mini-tasks that are intended to let you play around with the different kinds of loops. Please keep in mind that once again there are multiple ways to solve each subtask.

a) Write a program `loop1.cpp` that inputs an `unsigned int` from the user, then keeps subtracting 1000 from this number until it is smaller than 1000, and finally outputs the reduced number. You are not allowed to use the modulo-operator[1] here.

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
123456
456
```

Submission: https://codeboard.ethz.ch/ifmp16E4T2a

b) Write a program `loop2.cpp` that inputs an `unsigned int n`, and then outputs $\sum_{i=1}^{n} i^2$

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
10
385
```

Submission: https://codeboard.ethz.ch/ifmp16E4T2b

c) Write a program `loop3.cpp` in which you let the user enter `unsigned int`s until he inputs 0, and then output the number of inputs the user entered (including the 0).

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
1 2 1 2 0
5
```

Submission: https://codeboard.ethz.ch/ifmp16E4T2c

d) Write a program `loop4.cpp` that inputs an `unsigned int n` and then outputs the following star-triangle: on the first line print 1 star, on the second line print 2 stars, ..., on the nth line print n stars.

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

[1] This loop is a simple but inefficient way of simulating the modulo operation using subtraction. Actually, this is exactly what the Euclidean algorithm for computing greatest common divisors does (see Lecture 1); only the modern version of the algorithm employs the modulo operation, since programming languages usually support it. In Euclid's days, only subtraction was an operation that could easily be done manually.

```
5
*
**
***
****
*****
```

Submission: https://codeboard.ethz.ch/ifmp16E4T2d

Assignment 3 – Counting Divisors (4 points)

Write a program `kdivisors.cpp` that inputs a natural number `k` (including `0`) and outputs a list of all numbers `n` between 1 and 1000 with exactly `k` divisors (including `1` and `n`).

For example: the number 16 has 5 divisors: 1, 2, 4, 8, 16

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
5
16 81 625
```

Submission: https://codeboard.ethz.ch/ifmp16E4T3

Assignment 4 – Approximating π (4 points)

[Skript-Aufgabe 69]

The number π can be defined through various infinite sums. Here are two of them.

$$\begin{aligned} \text{Formula 1:} \quad \frac{\pi}{4} &= 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots \\ \text{Formula 2:} \quad \frac{\pi}{2} &= 1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \cdots \end{aligned}$$

Write a program `pi.cpp` that inputs a number n and then computes an approximation of π for both formulas up to the first n terms each. Which formula is better for that purpose?

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
3
Formula 1: 3.46667
Formula 2: 2.93333
```

Submission: https://codeboard.ethz.ch/ifmp16E4T4

Challenge - n-Queens-Problem (8 points) [Skript-Aufgabe 55]

Are you a chess player? Do you like mathematical considerations? Then look no further, this challenge is for YOU! This week's challenge is exercise 55 from the script.

To implement your program, please refer to the description on how you can write your own programs. To test your program you can try whether it computes that 2 solutions exist for the 4-queens problem and that 10 solutions exist for the 5-queens problem.