

## **Assignment 3**

*Dushan Terzikj*

*24.02.2017*

### **Problem 1:**

(a) Methods are implemented in the following source files:

- (i) 'Fib\_naive.cpp' → naive approach with STDIN and STDOUT
- (ii) 'Fib\_naive\_generator.cpp' → naive approach with generating increasing N
- (iii) 'Fib\_bottom\_up.cpp' → Bottom up approach with STDIN and STDOUT
- (iv) 'Fib\_bottom\_up\_generator.cpp' → Bottom up approach with generating increasing N
- (v) 'Fib\_matrix.cpp' → Matrix approach with STDIN and STDOUT
- (vi) 'Fib\_matrix\_generator.cpp' → Matrix approach with generating increasing N
- (vii) 'Fib\_closed\_form.cpp' → Closed form approach with STDIN and STDOUT
- (viii) 'Fib\_closed\_form\_generator.cpp' → Closed form approach with generating increasing N

There is a 'Makefile', which you can run using the command line by typing 'make' in the directory of the source files and the 'Makefile'. Then run the programs by typing `./[name_of_the_program]` in the command line (**NOTE:** Generator programs might take a while, they were initially used for generating run-time data).

(b) Table can be found in 'table.pdf'

(c) In practice yes. However, if not properly regulated, **closed form approach** might not give the correct n-th fibonacci number. The reason for this is floating point precision. In my program, I used round(x) function from C++ library <cmath>.

(d) Plot can be found in 'plot.pdf'.