Programming 2 – Laboratory 3

Part 1 (1 point)

Using provided files (*main.cpp* and *Sequence.h*) implement in separate file a definition of the constructor for *Sequence* class declared in *Sequence.h*. Class *Sequence* stores length and array of terms. Define a method *length()* which return a length of sequence.

Part 2 (2 point)

Overload needed operators for printing sequence and setting its terms.

Part 3 (1 point)

Implement method which get and set value of specified term in Sequence..

Part 4 (1 point)

Write method *isArithmetic()* which check whether terms in Sequence constitute the arithmetic sequence. Write function *check()* which takes as parameter a Sequence object and print information whether is arithmetic sequence.

Part 5 (1 points)

Define function *nth()* which create a new sequence form every n-th term of given *Sequence*.

```
=== PART 1 ===
Lengths of sequences :
c1 = 0
c2 = 5
c3 = 10
c4 = 10
c5 = 10

=== PART 2 ===
- c1 (default) []
- c2 (1 argument) [0 0 0 0 0 ]
- c3 (2 arguments) [1 2 3 4 5 6 7 8 9 10 ]
- c4 [1 2 3 4 5 6 7 8 9 10 ]
- c5 [1 2 3 4 5 6 7 8 9 10 ]
```

```
Input sequence
Give length : 5
Input terms : 9 3 2 6 3
Your sequence : [9 3 2 6 3 ]

=== PART 3 ===

Input term index (0-10) : 5
Value c3[5] = 6
Input new value c3[5] = 0
Value c3[5] = 0

=== PART 4 ===

Is a sequence c = [1 2 3 4 5 0 7 8 9 10 ] is arithmetic?
Function checkArithmetic: is NOT arithmetic.

=== PART 5 ===

Select every n-ty term from sequence [1 2 3 4 5 0 7 8 9 10 ]
Input n : 2
Result : [1 3 5 7 9 ]
```