**Lecture test (test4) in course "Algorithms and programming"  
May 18, 2021 @18:00**

**The procedure.**

1) Assignments are published in this page.

2) Time 90 minutes (until 19:40, including extra time buffer) to submit the solution in this page.

3) Tasks should be filled either in file/files (e.g., txt, docx) or on paper and then scanned (e.g., pdf).

4) Solution of each task should be put in a separate file (or files; multiple files per task can appear if solutions are written on paper and then scanned), so at least 4 files should be submitted for the full answer.

5) File names of solutions should start with the task ID (including both the letter and the digit, if applicable), e.g., A.docx, B.pdf, C2.cpp, D2.py.

6) Consultations during the test in MS Teams.

*Solve* ***one*** *task of each task group (where possible, only one mark per task group constitute the final grade, maximum amount of points to be awarded for particular task is given in square brackets after task number).*

**A[9]. Describe the notion: C-style strings vs C++-strings.**

**B. Write a C++ program to read from the console and to process a character string (char\* or string):**

B1[6]. Calculate the difference between amounts of even digits and odd digits. Example #1. s=“a12b34c”, answer=0 (both have 2 occurrences). Example #2. s=“a12b345”, answer=-1. Example #3. s=“a12b3468”, answer=2.

B2[9]. Calculate whether the character string s contains at least two occurrences of equal adjacent characters. Example #1. s=“a12**bb**34c”, answer=no (only ‘bb’). Example #2. s=“a12**bb**34**ccc**”, answer=yes (even 3: bb,cc,cc).

**C. Design a C++ function and provide the context how it is called to process a character string:** function ***removeFirstDigit*** to remove the first digit from the left found in the string, e.g., “x**2**3a45bc6”->“x3a45bc6”.

C1[6]. C++ string used.

C2[9]. C-style string (char\*) (used in-place (input overwritten by output – in the same string)).

**D[9]. Design a C++ class (with all fields defined as private) according to the source code that uses the class and the appropriate comments (from here you should figure out the list of required methods, amount of parameters for each, and its functionality). At least one of the methods should be implemented outside class.**

Class ***librarybook***:

librarybook b(“War and Peace”,1869,3);

// title, year, copies available

b.print(); // War and Peace 1869 3

// add more copies, check for a positive number, otherwise ignore

b.collect(2);

b.print(); // War and Peace 1869 5

// remove copies, check for a positive number and for having enough copies to remove, otherwise ignore

b.remove(1);

b.print(); // War and Peace 1869 4

// same title, year and copies specified as new values

librarybook c(b,1870,9);

c.print(); // War and Peace 1870 9