# C++ Advanced – Exam 2 (21 Apr 2019)

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++11 standard.

Submit your solutions here: <https://judge.softuni.bg/Contests/1608/CPlusPlus-Advanced-Retake-21-Apr-2019>

Any code files that are part of the task are provided under the folder **Skeleton**.

Please follow the exact instructions on uploading the solutions for each task.

# Task 4 – Overloading Madness

You are given 2 files: main.cpp and Matrix.h.

Your task is to study the provided Skeleton and implement the missing functionalities for Matrix.cpp.

As the name states the Matrix class is a representation of a simple 2D array of integers.

You need to implement the overloading of 4 math operations ‘add’, ‘subtract’, ‘multiply’, ‘divide’ and an additional overload for operator<<, which will print to the standard output (the console).

For the example let’s assume we have 2x2 Matrix A == Matrix B == |3 3|  
 |3 3|

* A += B would result in |6 6| A -= B would result in |0 0|   
   |6 6| |0 0|
* A \*= B would result in |9 9| A /= B would result in |1 1|   
   |9 9| |1 1|

**Important note**: remember than in math you can not divide by 0! Otherwise the universe would explode. If there seems to be any case of division by 0 -> simply treat the result as 0.

Example 5 / 0 = 0.

Keep in mind that the matrix sizes will **NOT** always be the same. In this case - simply perform the operation on their **common intersection** (the smaller matrix). You are **assured** that in this case the bigger matrix in size will be from the **LEFT** side of the mathematical operand (will simply be from the left).

For the example let’s assume we have 3x3 Matrix A |1 2 3| and Matrix B |4 3|  
 |4 5 6| |1 0|  
 |7 8 9|

* A += B would result in |5 5 3|  
   |5 5 6|   
   |7 8 9|

For the operator<< - print **whitespace** separated all element of matrix and of the end of each row print a **newline**.

Note: there is **whitespace** even after the last element on each row, before the **newline**.

You are given the main() function, which read from the standard input and populates MATRICES\_COUNT number of Matrices (each Matrix may have a different size and will always be a **square**(number of rows == number of cols)).

The matrices are then sorted (by their sizes) in Descending order (or at least their indexes).

Next 5 actions will be:

* the ‘first’ Matrix is added with the ‘second’ Matrix;
* the ‘first’ Matrix is subtracted with the ‘third’ Matrix;
* the ‘first’ Matrix is multiplied with the ‘fourth’ Matrix;
* the ‘first’ Matrix is divided with the ‘fifth’ Matrix;
* the ‘first’ Matrix is printed to the standard output (the console);

Your task is to study the code and implement the function so that the code accomplishes the task described.

You should submit a single .zip file for this task, containing **ONLY** the files you created.

The Judge system has a copy of the other files and will compile them, along with your file, in the same directory.

### Restrictions

Mathematical division by 0 (zero) is not allowed. Handle this case as explained above.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 5  5 5  2  4 4  4 4  2  3 3  3 3  1  2  1  5 | 2 6  6 6 |
| 2  5 5  5 5  2  4 4  4 4  2  3 3  3 3  2  2 2  2 2  2  5 20  20 0 | 2 0  0 0 |
| 1  10  2  1 2  3 4  3  9 8 7  6 5 4  3 2 1  4  2 2 2 2  2 2 2 2  2 2 2 2  2 2 2 2  5  9 9 9 9 9  9 9 9 9 9  9 9 9 9 9  9 9 9 9 9  9 9 9 9 9 | 0 6 4 11 9  15 24 7 11 9  8 9 10 11 9  11 11 11 11 9  9 9 9 9 9 |