

C++ Advanced – Exam 1 (07 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/1441/CPlusPlus-Advanced-Exam-07-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 1 – Memory Allocator

Your task is to write a simple memory allocator, which is not introducing a memory leak. You are given the **main()** function, which reads two values (as integer numbers) of memory followed by **N** command lines.

- The first integer value indicates the size for your memory allocator (in the range **[0, INT_MAX]** inclusive).
- The second integer value indicates the number of following command lines (**N**) you need to process and execute (in the range **[0, INT_MAX]** inclusive).
- The next **N** lines of indicate the command that you should process and execute.

The commands have the following syntax:

- "Allocate INDEX";
- "Deallocate INDEX";
- "Idle";

Where INDEX can be any integer in the range (in the range **[0, INT_MAX]** inclusive);

Keep in mind that INDEX may not be in present in your memory allocator boundaries.

You should implement the functions **executeCommand()** and **printResult()** in another .cpp file. (For example MemoryAllocator.cpp)

For each executed command in the **executeCommand()** – you should print a status message depending on the received **ErrorCode** in **printResult()**. Every call to **printResult()** should end with a **newline**.

You should print:

- For successful allocation/deallocation (not introducing memory leak or crashing the problem) – "**command** - success"
 - For preventing a memory leak – "**command** - memory leak prevented, will not make allocation"
 - For preventing a system crash – "**command** - system crash prevented, will skip this deallocation"
 - For receiving an index that is not in the bound of your memory allocator – "**command** – out of bound"
 - For receiving an "Idle" – "**command** - this exam is a piece of cake! Where is the OOP already?!?"
- Where "**command**" is the exact same string that is passed to the function.

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

You are free to implement another function/functions that are used internally by the **executeCommand()** and **printResult()**.

Examples

Input	Output
5 2 Allocate 3 Deallocate 3	Allocate 3 - success Deallocate 3 - success
2 4 Deallocate 21 Allocate 1 Idle Allocate 1	Deallocate 21 - out of bound Allocate 1 - success Idle - this exam is a piece of cake! Where is the OOP already?!? Allocate 1 - memory leak prevented, will not make allocation
8 4 Allocate 2 Deallocate 2 Deallocate 2 Allocate 2	Allocate 2 - success Deallocate 2 - success Deallocate 2 - system crash prevented, will skip this deallocation Allocate 2 - success