# 02. Memory Management – Homework Exercises

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

Code should compile under the C++03 or the C++11 standard.

Submit your solutions here: <https://judge.softuni.bg/Contests/1153/03-Exercises-on-Pointers-References-and-Dynamic-Memory> (select “Compete” when prompted)

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 3 – Duplicates

You are given code that reads information about companies – each having a **name** and a **list of employees by their initials** – and removes duplicate companies (companies with the same name are considered duplicates). The given code accomplishes this by using a function named removeDuplicates, which accepts a single parameter – a list of Company\*.

Note that the list of pointers may contain multiple pointers pointing to the same Company object, as well as pointers pointing to different objects, which have the same name. The function should remove any duplicates but leave their first occurrence in the list (i.e. the first time a Company is found in the list, it is considered the “original” and all subsequent Companies with the same name are considered duplicates).

Also note that you should ensure that removed duplicates are cleared from memory, using the delete keyword.

You should submit a single .zip file for this task, containing ONLY the RemoveDuplicates.h file, containing an int main() function that solves the task described.

|  |
| --- |
| **RemoveDuplicates.h** |
| #ifndef REMOVE\_DUPLICATES\_H  #define REMOVE\_DUPLICATES\_H  #include "Company.h"  // Place your code here  #endif // !REMOVE\_DUPLICATES\_H |

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

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Explanation** |
| uni (I.K.,S.N.)  \*begin  \*end  \*begin  joro (G.G.)  \*begin  \*begin  end | joro (G.G.)  uni (I.K.,S.N.) | The \*begin line means insert a pointer to the last company at the start of the list. The \*end line means insert such a pointer at the end of the list. Hence, after reading the input, the program will have the following list (\* denotes an object inserted as a copied pointer):  \*joro (G.G.)  \*joro (G.G.)  \*uni (I.K.,S.N.)  \*uni (I.K.,S.N.)  uni (I.K.,S.N.)  \*uni (I.K.,S.N.)  joro (G.G.)  In this list, the joro company is followed by the uni company, even though it is created later – that’s due to the inserting of the \*begin pointers after creating the joro company. Hence joro will appear first in the output, and uni second, because the first pointer to joro in the list is before the first pointer to uni in the list.  The idea behind this unusual input is that your code for the removeDuplicates function should not assume anything about the input – just traverse the list and remove the duplicates, don’t attempt to figure out a way to cheat the input |