



Programação

2020/2021

Project 4

Data limite de entrega: 12 de abril, 23h:59m; Peso de 1 valor em 20 valores na nota final

Description

A company has a list containing the identification of employees who have been designated as “Employee of the Month” throughout its various months of operation. However, this list is not coherent, sometimes using names and other times using employee numbers (note: the list does not contain both the name and employee number of the same individual, and there are no employees with the same name). An employee can be “Employee of the Month” more than once. When more than one employee is considered “Employee of the Month,” their information should be included in a sublist of the aforementioned list.

Develop a Python function that receives the list described above and returns two lists:

A list of lists, where each sublist contains the identifier (name or employee number) of each employee and the respective number of times they were considered “Employee of the Month”; this list should be ordered by the number of times they were considered “Employee of the Month” in descending order.

The original list (same id), without duplicates, containing the identifiers of the employees, ordered by the number of times they were considered “Employee of the Month” in descending order.

Example of the program output:

Original list: ["Manuel", 34567, "Manuel", 3425, 7865, "Alberto", 34567, ["Manuel", 3425, 7865], 8976]

Returned lists:

[["Manuel", 3], [34567, 2], [3425, 2], [7865, 2], ["Alberto", 1], [8976, 1]]

["Manuel", 34567, 3425, 7865, "Alberto", 8976]

Grading

The maximum grade for the practical work is 1 point (out of 20 points). Each group (two students) must develop a solution and submit it on Inforestudante by 11:59 PM on April 12, 2021. They must submit a .zip file containing a .py or .ipynb file and a .pdf of the code.

There are four levels of grading:

0 – the solution is incorrect;

0.5 - intermediate situation, i.e., a solution with a partially incorrect result, or with an incorrect result but an algorithm very close to the correct one;

0.75 - intermediate situation, i.e., a solution with a correct result, but with unoptimized code or that does not make adequate use of the materials taught in class;

1 - the solution is correct, with optimized code and makes appropriate use of the knowledge taught in class;

Defense rules:

each work is subject to defense, and each group member must be able to explain the solution and answer questions from the faculty;

the defenses take place in the last class dedicated to the topic, according to the course plan;

in online classes, students must have the camera and sound activated during the defense;

after the defense, the grade assigned to each student may be lower than the grade of the work, depending on the individual performance during the defense.