We have a group of employees who can achieve a certain amount of work, depending on his or her capability. We represent this amount of work as task\_point (a integer).

This group of employees needs to complete a list of tasks.

The task complexity is also represented by task\_point.

The harder the task, the higher the task\_point is.

The goal of this exercise is to find, using Python, the best possible way to dispatch tasks between all the employees so that the workload of each employee is as evenly spread as possible (so that one employee is not overworked, while another has almost nothing to do)

Every task has a name, and each employee has a nickname.

For each employee, the result should display the following as a minimum:

employee's nickname

employee's personal task\_point

the list of all tasks assigned to the employee (name and corresponding task\_point)

the total amount of task\_point assigned to the employee

You can structure the data the way you see fit. What's important is that they be defined in a way that is easy to update, so that we can run some tests.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Kindly send your solution by email and we will review it as soon as possible.

(please make your filename in this format : FirstName\_LastName.py)

We will be looking at code quality, methodology, approach of the problem and best practice. We are interested in seeing what you are capable of, so please feel free to take as much time as you need.