CenturyLink Cloud Coding Challenge

# Problem Statement

CenturyLink Cloud needs a small library that can be used as part of a larger program to calculate a department and/or manager’s monthly expense allocation based on the number and types of employees reporting to the manager or contained within the department. There are other parts of the system that deal with persistence and other related, but not core, features. This library should focus on solving the problem below using the appropriate design practices for the chosen language.

# Specific Requirements

* Developers warrant an allocation of $1000 each.
* QA Testers warrant an allocation of $500 each.
* Managers warrant an allocation of $300 each.
* Managers can have QA Testers, Developers and other managers report to them.
* Only managers can have people reporting to them.
* People will report to only a single direct manager.
* Departments consist of any kinds of employees.
* Users of this library should be able to:
  + Determine the monthly expense allocation warranted a manager who has various employee types reporting to him/her. The level of depth of the hierarchy should be flexible ideally.
  + Determine the monthly expense allocation warranted a department with various employee types under it at multiple levels deep just as the Manager can.

# Example Scenario

* For the given hierarchy:
  + Manager A
    - Manager B
      * Developer
      * QA Tester
* Manager A’s allocation should be: $2100

# Project Details & Constraints

* Please provide only a library as your solution. Our review will be based on quality and simplicity of code while still following good design principles, presence and quality of unit tests, and the discussion held in-person.
* The project should be zipped and emailed upon completion.
* Please use a modern functional or object-oriented language to solve this challenge. Examples would be Ruby, Python, C#, Java, go, javascript, erlang, Haskell, clojure, or similar language.