**Change Healthcare IMN Coding Challenge**

**Background**

There are many problems in programming that involve collections of objects that are very similar but differ in their containing ability. A common case is that of an employee hierarchy. All people in a company are employees, yet some are managers and so, in a sense, contain collections of other employees.

**Problem Statement**

For the following office hierarchy, write a test program that does these things:

* Read the input data (below) from a local file
* Print out an ASCII employee tree (any format you want)
* Print out the total salary requirements for the entire company.

Consider creating objects that would encapsulate the employee and manager properties.

**Extra Credit**: Sort employees alphabetically

**Mega Extra Credit**: Build unit tests for the code

**SuperMega Extra Credit**: Utilize language constructs, frameworks, or libraries to represent the data objects and perform automatic data validation on the data models.

Please use copious comments and document fully all your assumptions and reasoning. It’s better to have a clear, documented approach that doesn’t compile than to have something that works but is hacked up and undocumented.

**Input File Contents**

[

{

"id": 1,

"first\_name": "Dave",

"manager": 2,

"salary": 100000

},

{

"id": 2,

"first\_name": "Jeff",

"manager": null,

"salary": 110000

},

{

"id": 3,

"first\_name": "Andy",

"manager": 1,

"salary": 90000

},

{

"id": 4,

"first\_name": "Jason",

"manager": 1,

"salary": 80000

},

{

"id": 5,

"first\_name": "Dan",

"manager": 1,

"salary": 70000

},

{

"id": 6,

"first\_name": "Rick",

"manager": 1,

"salary": 60000

},

{

"id": 9,

"first\_name": "Suzanne",

"manager": 1,

"salary": 80000

}

]

**Sample output**

Jeff

Employees of: Jeff

Dave

Employees of: Dave

Andy

Dan

Jason

Rick

Suzanne

Total salary: 275000