Employee Profile Manager
CS 261 Final Project

Group 3
Christian Herinckx
Tieying Chu
Isaac Towai
Philip Warton

Files:

- makefile
- hashMap.c
- hashMap.h
- userInput.c
- userInput.h
- main.c

Structs and Datatypes:

```
Date {
        Int month
        Int day
        Int year
}

Employee {
        String first
        String last
        String id
        Date date_started
        Bool currently_employed
}
```

--- For internal use ---

- *HashLink*
- *HashMap*

Functions:

```
HashMap* read_employee_list();
Employee* create_new_employee();
Int prompt_user_choice();
Void employeePrint(Employee e);
Void employeeDelete(Employee e);
```

All hash map functions have been modified to accommodate employee data type instead of integer, and there are many other functions to accommodate the needs of our program which are not necessarily listed here.

User Experience:

The user can choose to either create a new profile to enter it into the system or to view a profile by entering the employee id. The user can then view all relevant information on that employee quickly and effectively.

This is implemented using a hash table for viewing efficiency.

Complexity:

To add a new employee will have a worst case scenario time complexity of O(n) due to the nature of hash tables, however to view profiles it will have an averaged amortized time complexity of O(1), with a worst case scenario of O(n).

File Format:

We expect to read from a pre-existing file to see the current list of employees before using the system, the format for this file is as follows:

```
N
FirstName LastName Idstring mm/dd/yyyy 0
FirstName LastName Idstring mm/dd/yyyy 1
...
FirstName LastName Idstring mm/dd/yyyy 0
```