Prelab 11 – Binary Search

For this prelab you will need to implement some functions that are specialized for use with Employee objects/structs, each of which has an integer member "SSN" for Social Security Number and integer "ID" for employee identification number. For the first function, the user will pass an array of pointers to N employee structs, along with the integer N, and the function will return a "database" that can be efficiently searched (returns NULL if it can't be created). The second function permits a pointer to an employee record to be efficiently retrieved based on a given SSN. The third function does the same except it efficiently retrieves based on a given employee ID. The last function is called when the user is finished with the database.

```
EmpDatabase createSearchableEmployeeDB(Employee **, int)
Employee * findEmpBySSN(int, EmpDatabase)
Employee * findEmpByID(int, EmpDatabase)
void freeEmpDatabase(EmpDatabase)
int getErrorCode(EmpDatabase)
```

The requirement that the two search functions must "efficiently" retrieve the requested employee record is to be interpreted as doing so in O(log(N)) time. That means you'll have to create *at least* one array with pointers to employee records that you have sorted by a particular member so that you can exploit the efficiency of binary search. If no employee record is found that satisfies a request, then NULL is returned.

The getErrorCode function can be invoked by the calling program/user to determine if the previously-called function produced an error. How can it do that? Note that this is a very different mechanism for providing error information to a user.

- JKU