Harford Community College Professor Brown

STEM Division Computer Science II

**Final Project**

Write a program that will simulate some of the functionality for an employee database file. Your database will need to execute the following functionality.

# Data:

Each record will consist of six attributes, one if with is the primary key. The primary key will be automatically assigned to a record when the record is created. The fields for the record are as follows:

**Field Name**: Description

**ID**: integer (automatically generated by program) **Last name**: 20 character max

**First** **Name**: 20 character max

**Social** **Security** **Number**: 11 character max (include “-“) **Salary**: float

**Age**: integer

The data should be stored in a randomly accessible file (binary file). There should be at least 20 records stored in the file.

# Index:

The index function will be able to index the records on any given field in ascending or descending order. The Index function should prompt the user to enter the field to be indexed. In addition, the program should prompt the user as if the indexing will be in ascending or descending order. To complete this task the sorting should be done using merge sort, heap sort or quick sort. Also, when you are indexing, you should not have moved full records within the database file . The whole record should not be read in the program. Only the primary key and the indexed field should be stored in the program’s memory.

# Query: Extra Credit

The query function will allow the user to perform basic queries by typing in SQL statements. This function should support the following SQL statements:

Select \* from table (basically it displays all records)

Select <field> from table (displays only the <field> listed from every record) Select \* from table where <field> = value (displays all records where <field> = value)

 Select <field> from table where <field> = value (displays only the <field> of all records where field = value)

# Display:

The display function should list all of the records in order based on the last indexing. In addition, the field that was indexed last should be the first field listed for each record. If no indexing was performed before choosing the display function, then the records should be displayed in order based on the primary key.

# Search:

The search feature should employ a search algorithm that performs at log base-2 or better. The search feature should prompt the user to enter a social security number. The program should print the record if one exists that matches the social security number, otherwise tell the user that the number is not in the system. If you are using the binary search method, I suggest you call the indexing method to sort the social security numbers

# Add Record:

This feature allows the users to add a record to the file. In addition, it should automatically generate a primary key value for the record. The program should insure that the new record has a unique social security number and that the format of the number is entered correctly (xxx-xx-xxxx)

# Delete Record:

This function will just delete a record from the file.

# File:

The file should store the records as a random access file. This will allow the program to access individual records without having to read the whole database into memory.

# Menu:

The menu function should display all of the abovementioned choices. Each function should be associated with an integer value. The program should prompt the user to enter in all of the choices at **once.** Then, the choices will be placed in a queue to track the order of the requests and executed one after another.