The objective of this homework was to gain a basic understanding about what the tasks of a Database Management system are and how that kind of program might go about doing its tasks. The operations that our program was to implement were 8, they were Create New Database, Open Database, Close Database, Display Record, Update Record, Create Report, add a record, and Delete a record. My program is laid out into two while loops one that checks if the user is still active and the other is to check if the user is still using the same database. Furthermore, I use a switch statement in order to keep track of what operation the user has chosen. In this report I will attempt to thoroughly explain the approach that I took to create this program by separating it into the 8 operations that were mentioned above.

**First While Loop:**

In this first while loop I did the operations that I felt were different from the operations that you could do to an instance of a database.

**Create A Database:**

The process that I used to create a new database was to ask the user for the name of the database and then I created a new file with that name and added the header line with the attribute values that we were given. I also appended a 0 to the end of this file which meant that there were no records in that file.

**Open Database:**

In this option I ask the user for input, meaning the filename, then I attempted to open the file. If the file did not exist, then I return an error message to the user prompting him to enter another filename.

**Second While Loop:**

In this second while loop was where I implemented the options that you could do to an instance of a database.

**Display Record:**

In order to display or search a record I used the binary search code that was given to us, but I made a couple of modifications. I first prompt the user for an id value after which I pad it with a function in order to be able to compare it to one that was pulled from the database. I then continue to do the binary search on the file. In the binary search function, I noticed that it was not retrieving the last tuples when the file got too big, after several hours of debugging I realized that the binary search was not taking into account that there were missing tuples and so when there was the last record would not be found because it would multiply the record size times the record and it would overshoot and end up with end of file flagged raised. What I did to solve this issue was that if during the binary search it overshot the value of the last tuple I retrieved the last valid Id that it searched and then did a linear search over the file. If the record is found in the binary search than it goes on to check if the experience value is “null” if it is, it goes on to claim that the record was not found. This is how I check if a searched file is deleted.

**Update Record:**

A record is updated using my program by first prompting the user for the record that they want to modify, I then search to see if the record exists. If the record does not exist then I give feedback to the user letting them know that the record does not exist, if however, it does exist then I output the record to the user, and I ask them what attribute they want to modify. After they have given me a value I check to see if the length is of the right size if not then I let them know that it is too long and to give me another value. After a value with a valid length is given then I calculate the displacement that I need to take to get to that attribute and after opening the file in read and write mode I insert the new value.

**Create Report:**

To create a report that outputs the first 10 records, I used a Linux command that eased the process. First I check to see if the file that the user has opened contains any records, if it does not then I let the user know that it is empty. If it does, I check to see how many records there are if there are less than 10 then I let the user know that the database contains less than 10 and then I use the sed Linux command to output all of the records in the database. If not, then I use the sed command to output the first 10 records in the database.

**Add a Record:**

To Add a record, I prompt the user for the values that they want to insert into the database, except for the ID value, that value I set myself by looking at what the last ID value is and then add 1. After I have checked the length of all the values to make sure they are not too long I then insert the record into a new file named temp, and I let the user know that the record was added.

**Delete a Record:**

To delete a record, I first do a binary search to check if the record exists if it does then I set the experience value to null which means that the record has been deleted.

**Merging Files:**

To merge the files, I used some Linux commands, first I create a file named NewHeader.txt where I store the header of the database along with how many rows are currently in the database this includes those that were added to the temp file when you create a new record. I then create a file named temp1.txt which I add all of the lines in the opened database excluding the header. I then merge and sort the records into another temporary file name systemtemp.txt. Finally, I merge the systemtemp.txt and NewHeader.txt file into the original file. I then deleted all of the intermediate files that I created.

In my program there are 4 places where the files can be merged. The first is when the user goes to search the file and there is a record that has been created in the temp file. The second is after creating 3 records. The third is when the user goes to close the database and there are some created records in the temp file. The fourth is when the user goes to create a report and there are less than 10 records in the files but there are some in the temp file.

**Compiling Code:**

In compiling my code, I had to add another library in order for it to work, the compiling code is inside the execute.bat file.

**Testing cases:**

In testing my code, I tried to think of cases where my code would mess up some of the test cases are the following:

* Input invalid options on the menus
* Input an attribute value that is too long
* Searching for a record that I just added
* Searching for the last records
* Creating a new database and then adding records
* Running a report on a newly created database
* Searching for a deleted record
* Search modified record

The file that I used is attached with this report.