**CSCE 4523 Database Management Systems**

**Homework 1**

**Due: Monday, February 1 at 11:59pm**

**By: Julio Sibrian**

**Objectives**

The objectives of this homework are to … implement a simple database using a sorted file of fixed length records. We needed to read from a data file and convert that data into our own formatted file which has fixed length records. The goal of this project is to understand and practice the use of file management techniques to implement a database system.

**Approach**

**How did you implement this? What languages? Strategies? Design?**

I implemented this project using object-oriented programming. I used c++ as my language as their file library is very straight forward and easy to use. I progressed incrementally creating and testing each class method one at a time to ensure it all ran smoothly. My design is object-oriented programming with classes.

**What format did you choose for your record? What size and order of fields?**

I chose to format my record with spaces since it was very easy to read records using c++ library <fstream> they automatically ignore white space so it was easy to retrieve information from my file. The size of the fields matched the max size of the csv file for example name had a max size of 83 characters. I ordered the fields the way I received them where id was first followed by region, state, code, name, type, and visitors.

**What delimiter? What total record size? Show a sample record**.

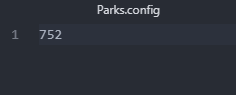
The delimiter for my .data file was spaces since it was easier to do so. My delimiter for reading the .csv file was the end line character because I found that manipulating a string was easier for me. The total record size is 152. Sample record:

6092 IM AZ HUTR Hubbell\_Trading\_Post\_National\_Historic\_Site National\_Historic\_Site 6197119

All are fields have their max size plus a space to separate the fields.

**What did you use the config file for? Show the config file.**

I only used the config file for storing the number of records in my .data file



**Did you use sample code?**

Yes, I used the c++ sample code.

How did you handle overflow file (methods)?

Every time there was an overflow, I would rewrite each record into a new file with blanks and the new record at its location with a blank following it as well.

**Results**

**Discuss Error handling, efficiency**

**What worked well? What didn’t?**

for error handling I have to make sure the record and its fields were the right size. If I didn’t then my file will not be a fixed length which will not allow me to use binary search. As with effciency my program is efficient because it uses binary search to find records which is way faster than searching for records linearly. Everything seemed to work fine. I had a lot of trouble with the record size because for some reason in windows \r has one more character, so my size was 152 for my computer and when I compiled in turing my size had to be 151 which took me a while to figure out.

**For pairs: who did what?**

I worked on this assignment by myself.

**Testing**

**Describe how you tested your program.**

**Refer to typescript: What worked, what didn’t**

**Discuss any additional test runs here.**

When making my program I tested each method individually starting with get record all the way to add / delete record which helped me understand my code better and solve my problems quicker.

I used the homework test cases for my typescript and everything worked. As for additional test runs I included them in my type script. I updated the first record and printed it out with display and create report. I also added extra records (where I displayed after adding) to show that my program can handle them.

**Typescript**

Include the typescript here (or upload it separately).

Included in the tar file.