Wilson Quon CCID: wquon CMPUT 291
Wenwen Li CCID: wenwen Mini Project 2
Jungu Liu CCID: jungu Davood Rafiei

The general overview of how our program will execute is split into the three phases explained in the requirements. In phase 1, the user can open the program from the terminal and they will be asked for the source xml file they wish to use. After this, the 4 text files will be created in the same directory. For phase 2, the user simply has to type the specified line given to create the index files. The index files will then sort and be created in the same directory.

In phase 3, the user is prompted to enter the query they wish to search for. After, a result will be printed out, showing the ad ID and the title of the ad. The user is then given an option to change the output by typing “of” (short for output=full) and once that is printed out, they can type “ob” (short for output=brief) to see the original output again. They can continually do this until they wish to quit by simply pressing “q”.

Important functions in our program:

get_queries(old_query): This is used to help separate each query individually and check the operators.

delet_space(string): This was used to get rid of any unneeded whitespace in the program. This does not delete the whitespace used to separate the queries. By deleting the whitespace, the user can type as many whitespaces they wish, but still get the same output.

get _*(***):** This represents the multiple functions used to for queries, checking for price, date, location, terms, and category. Each one is to check the user input and compare the files, then add them all to a list. The list is then returned after all the information is compared.

main(): Used for our input, changing from full to brief, and comparing our lists. The way our program will work is that it will first search through each query. Then, when comparing the queries. Each query will have a list, and the lists will compare their sizes with each other. The smallest one will be taken, and then the values that are the same will be returned into a new list, which will contain the final result.

Testing for the project started with phase 1. Initially, this went quite smoothly with the 10 ads file. However, as we used the larger files, we found more differences with our text files and the prepared ones. By using the “cmp” command in terminal, we were able to track our differences individually and see why they weren’t working. One of the main problems was St.-Alberta, and Banff-/-Canmore. Testing for phase 2 mainly came in phase 3, to see if they index file was accessible and processing the correct sorted information.For phase 3, we first initially tested for if we can get a single condition to work. For example, testing for if simply searching for a location will return the results we want. We created each test case as a separate function, splitting into location, price, categories, terms, and date. After this, we would check if our output would include all the correct values. While it is difficult to completely check for larger datasets, we checked first for every value from 10 records. We then checked if there was any incorrect values or ordering from 1000 records. After fixing a sorting error, we then proceeded to check if we could process two queries at once. We tried to come up with cases using each operator and for each function. After testing our own cases, we proceeded to check with the cases on the assignment page. Due to time, we did not test larger than the 1000 record files.

The group work was split relatively evenly for this project. For phases 1 and 2, since the total percentage of those 2 is 15%, Jungu and I worked on phase 1, while Wenwen used the prepared text files to complete phase 2. The time needed for phase 2 was relatively short, however for phase 1 it was much longer, due to the differences in text files as mentioned. Overall, phase 1 took about 5 hours to complete. For phase 3, we worked on this together in the lab rooms. This way, we were able to coordinate immediately with each other, allowing us to compare code, ask questions, and test results. Using a shared team drive on Google, we put our files on there so each group member has access. Since we spent much more time together as a group than project 1, our files were not nearly as confusing and much more coordinated. For phase 3, our combined time was around 16 hours. This final report was written by Wilson.

Assumptions: We are assuming the user will be inputting correct values when entering the data. This also includes the dashes and slashes with locations that have more than 1 word. However, the user is allowed to capitalize any letters they wish and input as many spaces they want. We are also assuming the file names shown in the specs will be the same as the ones used for testing.