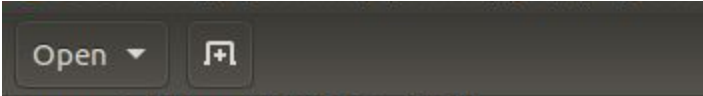**Project 1: Phone Directory (Sorting and Searching)**
Submit through Blackboard by 11:59pm, February 13.

1) Develop a telephone directory that consists of name and phone number pairs similar to the one in your smartphone with following functionalities. **[90%]**

- Your program should be able to create a new phone directory by reading from a text file. To do this, implement "**void create(text_file)**" function. In the text file, the name and phone number of a person should be separated by a comma, and there should be only one pair of a name and phone number per line. For example, see below:

- Once a phone directory is created as described above, sort all entries and display all the entries in alphabetical order of the First name. If multiple persons in the directory have the same First name, sort them based on their Last name. If they have the same first and last name, sort them based on their phone number. Use the randomized quicksort algorithm to do this. For this, implement "**void sort_contact()**" function.

- After displaying the sorted phone directory, search for a given name, e.g., Bob Smith. If found, return its location in the directory. Otherwise, return "Name not found" message and return. Use the binary search algorithm to do this. For this, implement "**void search(char* name)**" function.

- Implement "**void delete(char* name)**" function that searches for the name passed as the parameter, and delete the name and phone number found by search. If not found, print an appropriate error message and return.

- Implement "**void insert(char* name, char* phone_number)**" to insert a new pair of a name and phone number to the directory, keeping the directory sorted.

2) Coding style & Submission: [**10%**]

Write meaningful comments and function definitions. Make your code structured, indented, easy to read and robust.

Instructions for project submission:

1) Your code has to work correctly in ***remote.cs.binghamton.edu*** (e.g., remote01.cs.binghamton.edu) You can develop it in any environment as you wish but make sure you test it and run it in the ***retome.cs.binghamton.edu*** and package it from the same server before submitting. If your code does not work in the mentioned server you will receive no grade points for the respective algorithm. No exception will be considered.

2) Project name should be like this: *<language code>_<userid>_proj1.tar.gz,* all lowercase letters and tar.gzfile format
Example: **c_avora4_proj1.tar.gz** or **cpp_avora4_proj1.tar.gz**
Points will be deducted for bad project naming.

3) Your source code file name have to be exactly as below:
<user_id>_phone_directory.c (if you are using CPP then use .cpp extension instead of c)

4) Project should not containany any object, executable, readme, or input files. So, make sure you clean it up before submitting it. Points will be deducted for any presence of unwanted files. While testing, the TA will put **test_input.txt** into your project folder to test your projects.
So your project package will look like as below:
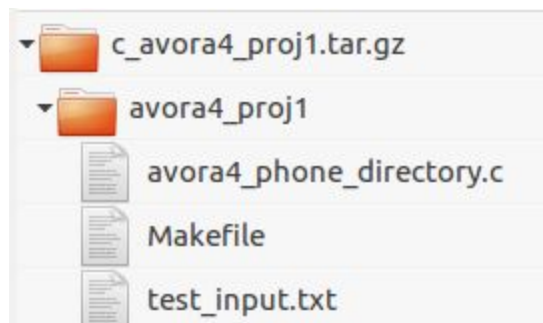
5) "**make**" command will **only** compile the source code files and it should not run any of the program. After executing "**make**" command it should create an object file with named **<userid>_phone_directory.out**

6) Your program should be executable by "**./<user_id>_phone_directory.out test_input.txt**"
If its not running by these commands then no credit will be given.

7) Make sure you don't look into others' code -- write code on your own. Copying from the Internet or any other online/offline sources is strictly prohibited. If automated tools find doing so, you will receive zero point for this project, and have to fill in and sign the official form that will be kept by the Binghamton University.

8) If BlackBoard says your submission is late it will be considered as late and 10% per day (including weekends and holidays) will be deducted. Even if you submit 1 second after the deadline, 10% will be deducted. So, please be timely.

9) We are running automated script and multiple tools to evaluate your code for different sets of inputs, so make sure you design your code well with appropriate loop choices. You don't need to implement any additional functions, loop joints, error correction, etc. unless you are asked to do so in the question above. If tools finds any unwanted implementation, they may fail some test cases which could reduce your points.

10) Contact the TA Ankur Vora (avora4@binghamton.edu) for any clarification needed.

_____

Important Policies

- All work must represent each individual student's own effort. If you show your code or any other part of your work to somebody else or copy or adapt somebody else's work found online or offline, you will get zero and be penalized per the Watson School Academic Honesty Code (http://www.binghamton.edu/watson/about/honesty-policy.pdf). To detect software plagiarism, everybody's code will be cross-checked using an automated tools. On the first violation, you will receive zero point for this project, and have to fill in and sign the official form that will be kept by the Binghamton University. On the second violation, it will be reported to the Watson School Committee for Academic Honesty for an official hearing.

- Your code will be compiled and executed. If your code does not compile or produce any runtime errors such as segmentation faults or bus errors, you will get zero. Before submitting your code, run it in a Linux machine (e.g., a remote machine) to remove any such errors. You can use "valgrind" (http://valgrind.org/) to debug memory errors, such as segmentation faults or bus errors.

- You are supposed to figure out how to implement this project. The instructor and TA will be more than happy to explain algorithms, such as quicksort or binary search; however, they will not teach you how to implement them or help you to make any kind of

decision. Neither will they read or debug your code. The instructor and TA will not take a look at an emailed code. If you have questions about general C/C++ programming or Linux, use Google search, which will be much faster than asking the TA. (Don't copy from the Internet though.)