**Summary of provided functions.**

1. Added function in BSTClass
   * Exact search (This do exact search and push to the vector)
   * Exact search sub tree ( This will be used for recursive call for above function)
   * Contain search (This do exact search and push to the vector)
   * Contain search sub tree ( This will be used for recursive call for above function)
   * Edit Node ( This change the contact record node and return pointer to node so it updated node can be changed in searched vector too)
   * Edit Affiliates (similar to edit Node)
   * Overloaded extraction operator ( ostream all the the record)
   * Output sub tree ( This is used for recursive for above function)
   * Print tree( Display short or in detail all the record on tree)
   * Print sub tree ( Recursive function for above)
   * Get root (return the root of the tree, helpful for recursive calls)
2. Functions in ContactsClass.cpp/h
   * Default Consturctor (create object with all empty field)
   * Setters
   * Getters
   * Overloaded extraction and insertion operator (do the extraction and insertion)
   * Overloaded smaller than and equal operator (based on ID for the tree Insert)
3. Functions in functions.cpp/h
   * Read from file (Load record from database)
   * Read from user file (Load record from user chosen database)
   * Write to database from tree (write to default database)
   * Write to user file from tree (write to the user chosen databas)
   * Search exact in tree (call the search exact function in BSTClass)
   * Search contain in tree (call the search contain function in BSTClass)
   * Search tree (combination above two function)
   * Search exact (search exact in vector)
   * Search contain (search contain in vector)
   * Search in search (combination of above two function)
   * Sort vector ( sort vector according to user need. Have option in body)
   * Save to file by field ( save the searched data from vector only requested field)
   * Save to file (save all field of searched data to user file)
   * Edit record (call edit Node/Affiliate on tree, also update the same record in vector)
   * Insert from user (create new record from user and insert in tree)
   * Print search (display searched result from vector, only short form)
   * Print from tree (calls print tree in BSTClass)
   * Searching editing and saving ( this is major function that call search functions, display them, redo search, edit them, save them to user file using functions in the functions.cpp)
   * Print contact ( display single record)
   * Print contact detail ( print all the record in detail using above function)
4. Starting menu.cpp
   * Initial display
     1. ( This displays the main menu and call functions as needed from functions.cpp)

**Design Document**

1. Title  
   Contacts Database, Bhuwan Sapkota, 5/8/16
2. Problem DescriptionThis program maintains the contacts database including adding, editing, reading and writing to the text files.

**Overall Software Architecture**This program will have following menus

1: Load database from default file.

2: Load database from new file.

3: Search and Edit the record

(This will include:

-Adding/Editing/Deleting affiliates from record.

-Editing record itself after search.

-Exporting the searched result to data file.).

4: Add a new contact in record.

5: Save database to default file.

6: Export database to new file.

7: Display all record from database.

0: Exit.

 (Design picture is in next page)

Untitled Diagram.png

1. Input Requirements

* Unique ID number. No input (program generated or read from file)
* First name (string)
* Middle name (or initial) (string)
* Last name (string)
* Company name   (string)
* Home phone (string)
* Office phone (string)
* Email (string)
* Mobile number (string)
* Street address (string)
* City (string)
* State (string)
* Zip code (string
* Country (string)

This will be same for all the affiliates and input from keyboard.

1. Output RequirementsThere will be two different data output in the file.
   * One is writing the data to the original data file or user chose data file from AVL tree (**These files will be usable to read the data to AVL tree in future**).
   * Other one will be user defined filtered contact list to the given filename with requested data field or all field of searched list. (**These files will not be usable to read the data to AVL tree in future**)
2. Problem Solution Discussion
   * Read the text file
   * Create ContactsClass objects
   * Create AVL tree;
   * Insert each object in the tree.
   * Ask user if they want to add more contacts and add it.
   * Ask user if they want to search based on fields (i.e. 1. First Name, 2. Last Name); ask them to choose the field. (I will be doing in-order search and push the value to the vector right away if found.)
   * Search result will be stored in the vectors of ContactClass.
   * If user wants to edit the contacts edit that in tree and update searched vector accordingly (user will not need to redo search)
   * Ask user what field they want use to sort the vector and sort using sortVector function.
   * Use sub searches in vector and override the search result in same vector so on
   * If user wants write them in the file write them.
   * Let the user choose whether/where they want to save the updated data(if any).
3. Data Structures
   * I decided to use vector over list is because it is less complex to access the elements.
   * Deleting the element especially in affiliates is same complexity in list and vector.
   * Sorting is same complexity because loop has to go iterate through whole container on both vector and list.
4. User Interface Scheme.

Welcome message

1. Load from the default file.
2. Load from user file.
3. Search contacts
   * Exact search
     + Edit contact
       - Just enter the displayed search result number.
     + Sub search (After search it will return to search contact sub menu)
     + Sort
       - Choose the sorting criteria
     + Write to file
       - Choose fields need to be written
         * Detail save
         * Selected field save

List of fields (with number for each field)

* + - * + Enter the file name to be saved.
  + Contain search
    - Edit contact
      * Just enter the displayed search result number.
    - Sub search (After search it will return to search contact sub menu)
    - Sort
      * Choose the sorting criteria
    - Write to file
      * Choose fields need to be written
        + Detail save
        + Selected field save

List of fields (with number for each field)

* + - * + Enter the file name to be saved.

1. Add new contact
   * Enter First name
   * Enter Last Name
   * ……// for all fields and affiliates.
2. Save Database to file,
3. Save Database to user file.
4. Display all the record in Tree
   * Short form
   * In detail.
5. Exit the program

**Status of the program**

1. This program is successfully compiled and should fulfill all the assignment requirements.
2. This is compiled in Xcode 7.3 and mac terminal successfully.

3. makefile is included that use c++11 flag.