OOP – Lab 1

BITF19 – Morning

Things That Will Get You Marks

Here are a few things that will get you marks easily:

- **Building Proper Header Files and Structure** – You are provided with the file structure, follow them closely and you'll easily earn maximum marks.
- **Commenting** – There is no comment-less code that a professional can easily understand. So, unless you want me to question your coding methodology and logic, properly comment the code and save yourself by making me understand your logic by words rather than code. Of course, I will find errors.
- **Proper Programming Conventions** – Follow all programming conventions taught during the PF course.
- **Pay Focus to the Logic** – This should go without saying. Your task is as easy as you make it. Don't ruin it by making it too complex.
- **Think About All Test Cases** – Pay special focus on test cases. Will your code run on all of them?
- **Make Your Code Unique** – Use good programming conventions, beautify the code, make good variables. Make your code clean and to the point. ***Plagiarism will not be tolerated in any case.***
- **Timely submit your assignment** – From this moment on, I will not allow anyone to submit the code late. If you are absent, turned the assignment late, forgot to turn in, or any other excuse, it's on you. I will not evaluate the code. If someone has emergency, get your application approved through Professor.
- **Follow File Submission Format** – I will be testing automated checking on this lab. All of you ***MUST*** follow this File Format. Anyone not following this may risk his evaluation.
    - **Folder** – Your folder should contain four files **QuotationDBOperations.h**, **QuotationDBOperations.cpp, QuotationDBApplication.h** and **QuotationDBApplication.cpp.**
    - **Folder Name** – Your Folder should be named LAB01_BITF19M0XX (Your Roll Number comes in place of XX)
    - **Zip Folder** – Your Folder should be compressed and zipped in a **.ZIP** file with the same name as folder.

Project Quotation Database

You are to design an application, which will store sayings/quotations of different renowned philosophers/writers etc.

This application should support the following operations.

- Let the user search the quotation on the basis of different keywords.
- Let the user search the quotation on the basis of author/writer name.
- It allows the user to add quotes along with author name (if author is not known then name it as anonymous).

For the above application you need to have following data structure that will represent/store the quotation data and its authors:

- char * * quoteList;
  pointer to an array of char pointers who's each location points to a quotation (array of characters)
- int numOfQuotations;
  stores the number of quotations that are currently stored in quotation database (the same we did in Set application for number of elements).
- char * * authorList;
  pointer to an array of char pointers whose each location points to an author name (array of characters). So, the quotation at quote[i] has an author at author[i]
- int capacity;
  size of array pointed by quote ***which should be resizable*** if database gets full.

Supported Operations:

1 **-** *void createQuotationDatabase (char * * * quoteList, char * * * authorList, int capacity);*

Receives the initial capacity of quote array

2 **-** *void addQuotation(char * * * quoteList, char * * & authorList, int noq, int * capacity, char * quote, char * author);*

3 **-** *void displayAuthorWise (char * * & quoteList, char * * & authorList, int noq, char * author);*
receives author name and display all the quotation by the author on console.

4 **-** *void displayQuotation (char * * & quoteList, char * * & authorList, int noq, char * quote);*
display on console all those quotation(s) containing the received string/word.

5 *- void removeQuotation (char * * & quoteList, char * * & authorList, int & noq, char * word);*
it displays all the quotations containing the received string/word and then ask the user for the quotation number on console which he wants to delete.

6 *- void freeQuotationDatabase (char * * * quoteList, char * * & authorList, int cap);*
Free the resources i.e heap captured by the data structure for quotation database.

Sample Run and Total Structure of Project and Code Distribution

QuotationDBOperations.h

```
void createQuotationDatabase (char * * * quoteList, char * * * authorList, int capacity);
void addQuotation(char * * * quoteList, char * * & authorList, int * noq, int * capacity, char * qt, char * at);
void displayAuthorWise (char * * & quoteList, char * * & authorList, int noq, char * at);
void displayQuotation (char * * & quoteList, char * * & authorList, int noq, char * qt);
void removeQuotation (char * * & quoteList, char * * & authorList, int & noq, char *);
void freeQuotationDatabase (char * * * quoteList, char * * & authorList, int cap);
```

QuotationDBOperations.cpp

```
/*
will contain the definitions of all function in 'QuotationDBOperations.h'
*/
```

QuotationDBApplication.h

```
void quotationDBApplication();
```

QuotationDBApplication.cpp

```
void quotationDBApplication() {

        char * * quoteList=0;
        char * * authorList=0;
        int numOfQuotations = 0;
        int capacity = 10;
        createQuotationDatabase(&quoteList, &authorList, capacity);
        addQuotation(&quoteList, authorList, &numOfQuotations, &capacity,"Love Pakistan","Nayar
        Ali");
        addQuotation(&quoteList, authorList, &numOfQuotations, &capacity,"Love Pakistan","Nayar
        Ali");
        addQuotation(&quoteList, authorList, &numOfQuotations, &capacity,"Be Honest","Asif");
        addQuotation(&quoteList, authorList, &numOfQuotations, &capacity,"Work Work and
        Work","Quad e Azam");

}
```

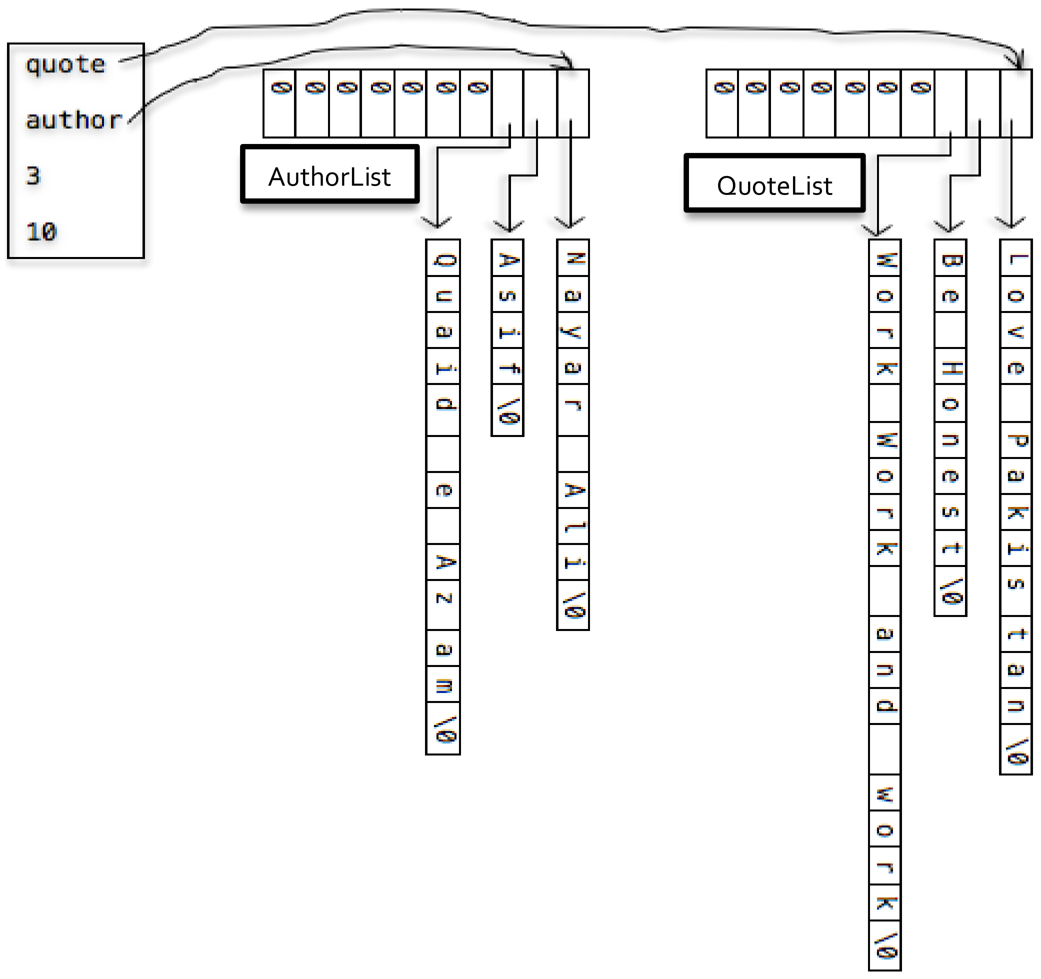
main.cpp

```
int main() {

        quotationDBApplication();
        return 0;

}
```

P.S. If You can’t understand the project structure, follow this diagram below.


quote
author
3
10
AuthorList
QuoteList
0 0 0 0 0 0 0
0 0 0 0 0 0 0
Quaid e Azam\0
Asif\0
Nayar Ali\0
Work Work and work\0
Be Honest\0
Love Pakistan\0