Objective:

- Dealing with Character Arrays and Heap
- Dealing with Array of struct on heap

Challenge-A: ***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 structs that will represent/store the quotation data and its authors:

```
struct Quote
{
    char * message;
    char * author;
};
```

```
struct QuoteDatabase
{
    Quote * data;
    int numOfQuotations;
    int capacity;
};
```

Supported Operations for Quote:

1. `void initializeQuote(Quote & q, const char * qt, const char * auth);`
   Initialize the quote q with given quote 'qt' and given author 'auth'.

2. `Quote createClone(Quote &q);`
   create deap copy of q.

3. `void freeQuote(Quote &q);`
   free the resources/heap captured by q.

Supported Operations for QuoteDatabase:

1. `void createQuotationDatabase (QuoteDatabase & quoteDB, const int capacity);`
   Receives the initial capacity for the quote array (array of objects of Quote struct on heap) and initializes the QuoteDatabase accordingly.

2. `void addQuotation(QuoteDatabase & quoteDB, const char * quote, const char * author);`

3. `void displayAuthorWise (QuoteDatabase & quoteDB, const char * author);`
   receives author name and display all the quotation by the author on console.

4. `void displayQuotation (QuoteDatabase & quoteDB, const char * quote);`
   display on console all those quotation(s) containing the received string/word.

5. `void removeQuotation (QuoteDatabase & quoteDB, const 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 (QuoteDatabase & quoteDB);`
   Free the resources i.e heap captured by the data structure for quotation database.

© Fareed Ul Hassan Baig

7. `void resize(QuoteDatabase & quoteDB);`
   While adding quotation if the database gets full then it resizes the QuoteDB automatically to double size without asking/bothering the user about new size.

Sample Run and Total Structure of Project and Code Distribution

Quote.h

```
struct Quote
{
    char * message;
    char * auhtor;
};
```

Quote.cpp

```
void initializeQuote(Quote & q, const char * qt, const char * auth);
{
    q.message = new char[getStringLength(qt)+1];
    q.auhtor = new char[getStringLength(auth)+1];
    stringCopy(q.message, qt);
    stringCopy(q.author, auth);
}
Quote createClone(Quote &q);
void freeQuote(Quote &q);
```

QuoteDatabase.h

```
struct QuoteDatabase
{
    Quote * data;
    int numOfQuotations;
    int capacity;
};
```

QuoteDatabase.cpp

```
void createQuotationDatabase (QuoteDatabase & quoteDB, const int capacity);
void addQuotation(QuoteDatabase & quoteDB, const char * quote, const char * author);
{
    if (isFull(quoteDB))
        resize(quoteDB);
    initializeQuote(quoteDB.data[quoteDB.numOfQuotations], quote, author);
    quoteDB.numOfQuotations = quoteDB.numOfQuotations + 1;
}
void displayAuthorWise (QuoteDatabase & quoteDB, const char * author);
void displayQuotation (QuoteDatabase & quoteDB, const char * quote);
void removeQuotation (QuoteDatabase & quoteDB, const char * word);
void freeQuotationDatabase (QuoteDatabase & quoteDB);
void resize(QuoteDatabase & quoteDB);
```

QuoteApplication.h

```
void quotationDBApplication();
```

QuoteApplication.cpp

```
void quotationDBApplication()
{
    QuoteDatabase quoteDB;
    createQuotationDatabase(quoteDB, 10);
    int choice;
```


© Fareed Ul Hassan Baig

```
    do
    {
        cout<<"------------Quotation Database------------\n";
        cout<<"******************************************\n";
        cout<<"Enter 1 to Add Quote \n";
        cout<<"Enter 2 to Remove Quote \n";
        cout<<"Enter 3 to Search Quotes of Author \n";
        cout<<"Enter 4 to Search Quote That Contains Word \n";
        cout<<"Enter 0 to Quit ";
        cin>>choice;
        cin.ignore();

        switch (choice)
        {
            case 1:
                char q[200], a[50];
                cout<<"\nEnter Quotation: ";
                cin.getline(q, 200);
                cout<<"\nEnter Author Name: ";
                cin.getline(a, 50);
                addQuotation(quoteDB,q,a);
                break;
            case 2:
                break;
            case 3:
                break;
            case 4:
                break;
            case 0:
                freeQuotationDatabase(quoteDB);
                break;
            default:
                cout<<"OOPs! You Entered Wrong Choice\n";
        }
    }
    while(choice!=0);
}
```

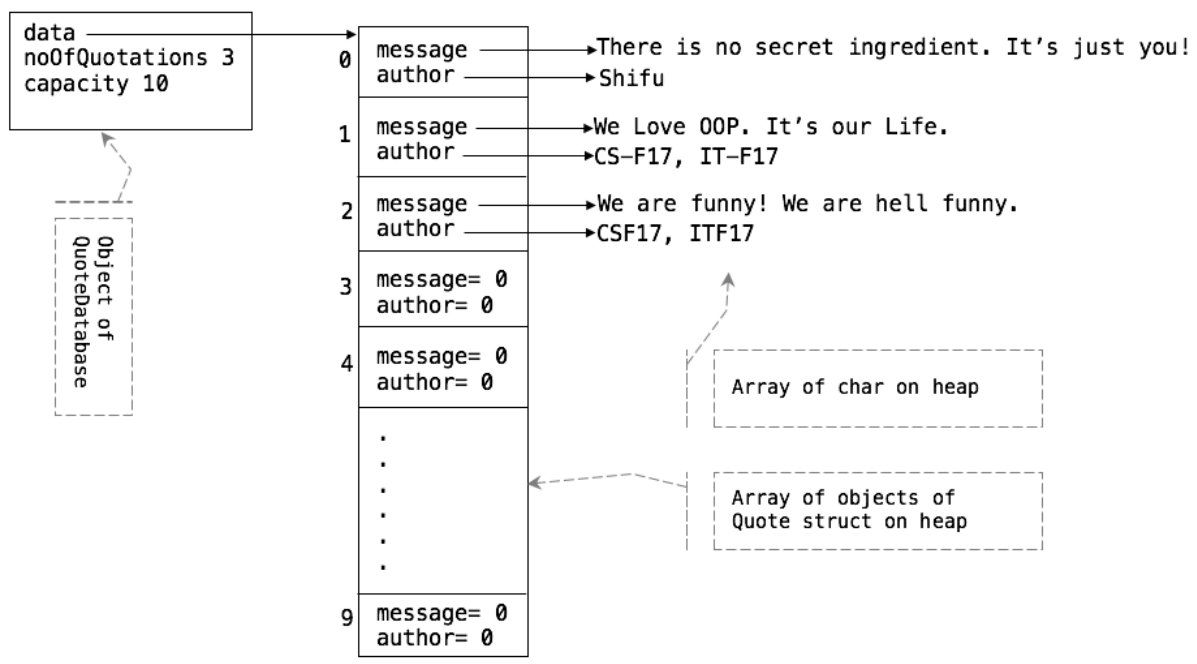
driver.cpp

```
int main()
{
    quotationDBApplication();

    return 0;
}
```


© Fareed Ul Hassan Baig

Sample Object Layout



How to Submit

- When you are done with the lab then create a folder and name it as your RollNo-Lab-3.
  - For example, roll no 1 of CS morning will name its folder as:
    - BCSF17M001-Lab-3
  - Put all your code files (.cpp and .h) in the above said folder.
  - Compress the created folder and send it as an attachment to the email: BSEF16A026@pucit.edu.pk
  - Reminder: Your Email body text must contain the oath by you as described in class.
- Your Email must reach to the said inbox positively on or before **01:30 PM** 22-Sep-2018.

THERE IS NO SECRET INGREDIENT.
IT'S JUST YOU.

© Fareed Ul Hassan Baig