



**Green University of Bangladesh**  
**Department of Computer Science and Engineering (CSE)**  
**Faculty of Sciences and Engineering**  
**Semester: (Spring, Year:2022), B.Sc. in CSE (Day)**

**Course Title: Structured Programming Lab**  
**Course Code: CSE 104                      Section:213DD**

**Lab Project Name: Phonebook System**

**Student Details**

Name		ID
1.	Md.Shaon Howlader	213902064

**Submission Date: \_\_\_\_\_15 May 2022 \_\_\_\_\_**

**Course Teacher's Name: \_\_\_\_\_**  
**Sultanul Islam Ovi**

**[For Teachers use only: [Don't Write Anything inside this box](#)]**

<b><u>Project Proposal Status</u></b>		
<b>Marks: .....</b>	<b>Signature: .....</b>	<b>Comments:</b>
<b>.....</b>	<b>Date: .....</b>	

# Table of Contents

## **Chapter 1: Introduction**

- 1.1 Introduction
- 1.2 Design Goals/Objectives

## **Chapter 2**

### **Implementation of the project**

- 2.2 Implementation
- 2.3 Screenshots

## **Chapter 3 Conclusion**

- 3.1 Learning Outcomes
- 3.2 Future Scopes

## **References**

# CHAPTER 1

## Introduction

### 1.1 Introduction:

C is a structured, high level machine independent language. C is converted to a lower language which was understood by the compiler. It allows the software developers to develop programs without worrying about the hardware platforms where they will be implemented. The C language comes from the ALGOL which gives the concept of structured programming to the computer science community. It was introduced early in the 1960's.

After, MARTIN RICHARDS DEVELOPED a language known as BCPL in 1967 for this in the 1970's Ken Thompson created a language from BCPL and he called it "B" both BCPL and B are typeless system programming languages. After finding ALGOL, BCPL, and B then from this C evolved from that at BELL LABORATORIES in 1972 by "DENIS RITCHIE". C Language uses many concepts from these and added the concept of data type because it was developed along with a UNIX operating system. UNIX is nothing but a most popular network operating system is used today and the heart of the internet data super highway.

### 1.2 Design Goals/Objectives

- To develop a "Phonebook contact" application using C programming.
- This program is very useful nowadays to store complete information under a single contact number.
- This program also has options to deletion and modification of the entered contact number.

# Chapter 2

## Implementation of the project

### 1.1 Flow Charts

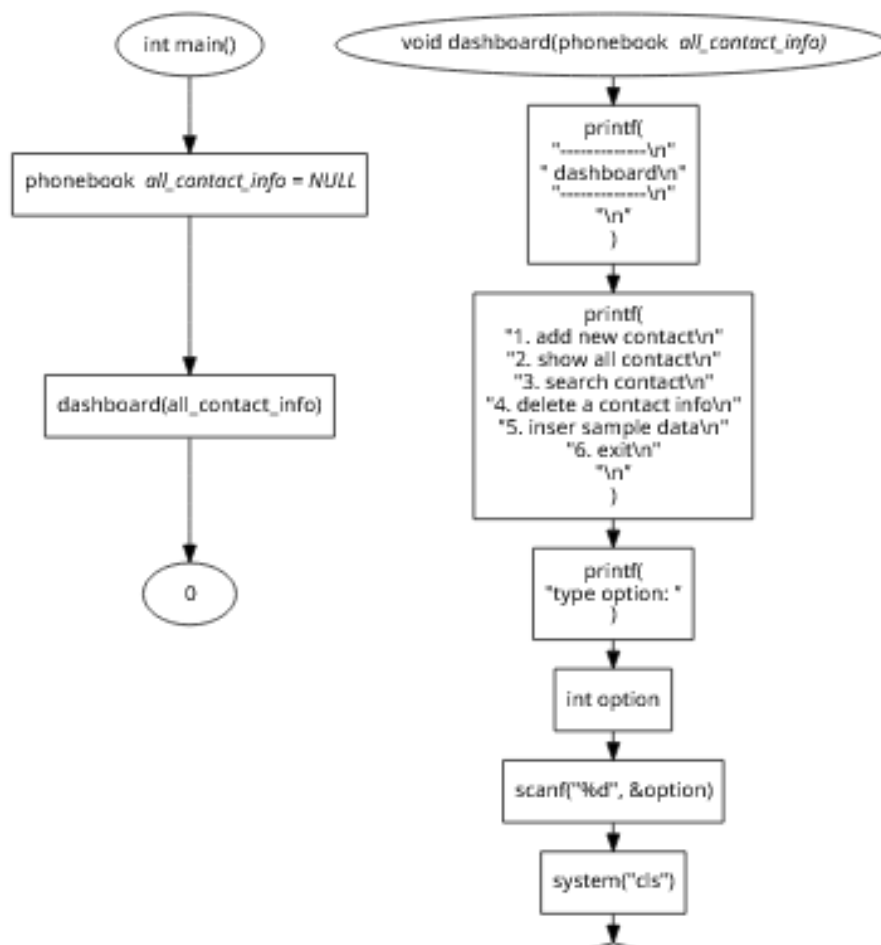


Figure 1: Flow Chart of Main Menu

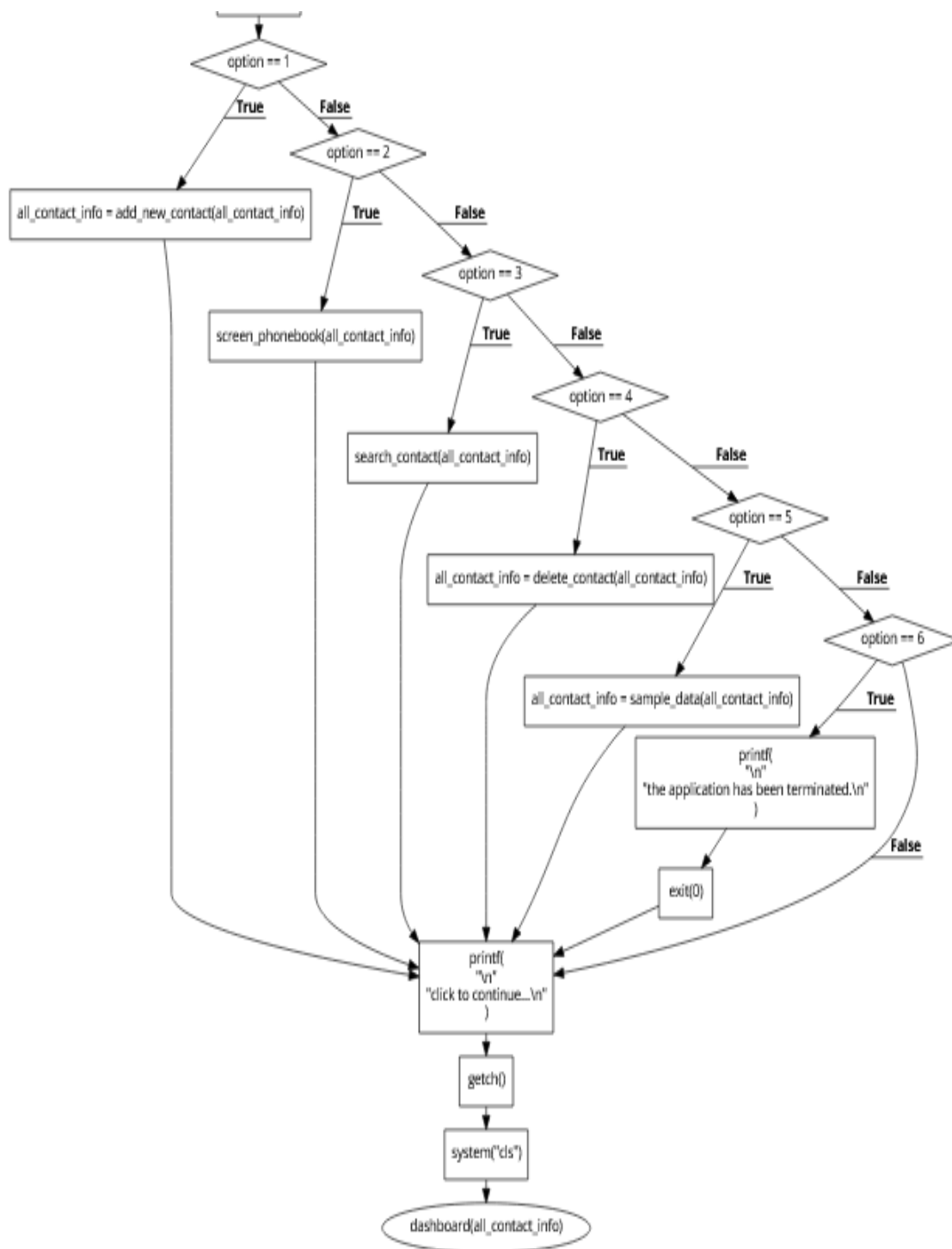


Figure 2: Flow Chart of different activities involved in different function of the project

## 2. Implementation

### C Source Code

```
/*
    Name: Md. Shaon Howlader
    ID: 213902064
    Email: shahriarhossenshaon4@gmail.com
    Project title: Phonebook System
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>

typedef struct contact_info{
    char name[20];
    char location[20];
    char email[20];
    char phone[20];
} contact_info;

typedef struct phonebook{
    contact_info * single_contact_info;
    struct phonebook * next_contact_info;
} phonebook;

phonebook * sample_data(phonebook * all_contact_info);
void dashboard();

contact_info * blank_contact_info();
phonebook * blank_phonebook();

void screen_phonebook(phonebook * all_contact_info);

contact_info * insert_contact_info(char * name, char * location, char *
email, char * phone);
phonebook * insert_phonebook(phonebook * all_contact_info, contact_info
* new_contact_info);

void search_phonebook_by_name(phonebook * all_contact_info, char *
name);
void search_phonebook_by_location(phonebook * all_contact_info, char *
```

```

location);
void search_phonebook_by_email(phonebook * all_contact_info, char *
email);

phonebook * delete_phonebook_by_name(phonebook * all_contact_info, char
* name);
phonebook * delete_phonebook_by_location(phonebook * all_contact_info,
char * location);
phonebook * delete_phonebook_by_email(phonebook * all_contact_info, char
* email);

phonebook * add_new_contact(phonebook * all_contact_info);
void search_contact(phonebook * all_contact_info);
phonebook * delete_contact(phonebook * all_contact_info);

int main(){
    phonebook * all_contact_info = NULL;
    dashboard(all_contact_info);

    return 0;
}

void dashboard(phonebook * all_contact_info){
    printf(
        "-----\n"
        "  dashboard\n"
        "-----\n"
        "\n"
    );

    printf(
        "1. add new contact\n"
        "2. show all contact\n"
        "3. search contact\n"
        "4. delete a contact info\n"
        "5. inser sample data\n"
        "6. exit\n"
        "\n"
    );

    printf(
        "type option: "
    );

    int option;

```

```

scanf("%d", &option);

system("cls");

if(option == 1){
    all_contact_info = add_new_contact(all_contact_info);
}else if(option == 2){
    screen_phonebook(all_contact_info);
}else if(option == 3){
    search_contact(all_contact_info);
}else if(option == 4){
    all_contact_info = delete_contact(all_contact_info);
}else if(option == 5){
    all_contact_info = sample_data(all_contact_info);
}else if(option == 6){
    printf(
        "\n"
        "the application has been terminated.\n"
    );
    exit(0);
}

printf(
    "\n"
    "click to continue...\n"
);
getch();
system("cls");

dashboard(all_contact_info);
}

phonebook * delete_contact(phonebook * all_contact_info){
    printf(
        "-----\n"
        " delete contact\n"
        "-----\n"
        "\n"
    );

    printf(
        "1. delete by name\n"
        "2. delete by email\n"
        "\n"
    );
};

```



```

    printf(
        "type option: "
    );

    int option;
    scanf("%d", &option);

    if(option == 1){
        char name[20];
        getchar();
        printf("type your name: ");
        gets(name);
        all_contact_info =
delete_phonebook_by_name(all_contact_info, name);
    }else if(option == 2){
        char email[20];
        getchar();
        printf("type your email: ");
        gets(email);
        all_contact_info =
delete_phonebook_by_email(all_contact_info, email);
    }
    return all_contact_info;
}

void search_contact(phonebook * all_contact_info){
    printf(
        "-----\n"
        " search contact\n"
        "-----\n"
        "\n"
    );

    printf(
        "1. search by name\n"
        "2. search by email\n"
        "\n"
    );

    printf(
        "type option: "
    );

    int option;
    scanf("%d", &option);

```

```

    if(option == 1){
        char name[20];
        getchar();
        printf("type your name: ");
        gets(name);
        search_phonebook_by_name(all_contact_info, name);
    }else if(option == 2){
        char email[20];
        getchar();
        printf("type your email: ");
        gets(email);
        search_phonebook_by_email(all_contact_info, email);
    }
}

phonebook * add_new_contact(phonebook * all_contact_info){
    printf(
        "-----\n"
        " add new contact\n"
        "-----\n"
        "\n"
    );

    char name[20];
    char location[20];
    char email[20];
    char phone[20];

    printf(
        "name: "
    );
    getchar();
    gets(name);

    printf(
        "location: "
    );
    gets(location);

    printf(
        "email: "
    );
    gets(email);

    printf(

```

```

        "phone number: "
    );
    gets(phone);

    contact_info * new_contact_info = insert_contact_info(name,
location, email, phone);
    all_contact_info = insert_phonebook(all_contact_info,
new_contact_info);
    return all_contact_info;
}

phonebook * delete_phonebook_by_location(phonebook * all_contact_info,
char * location){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    phonebook * blank = blank_phonebook();
    phonebook * temp1 = blank_phonebook();
    if(strcmp(each_contact_info -> single_contact_info -> location,
location) == 0){
        blank = each_contact_info -> next_contact_info;
        free(each_contact_info);
        all_contact_info = blank;
    }else{
        while(each_contact_info -> next_contact_info != NULL){
            if(strcmp(each_contact_info -> next_contact_info ->
single_contact_info -> location, location) == 0){
                each_contact_info -> next_contact_info =
each_contact_info -> next_contact_info -> next_contact_info;
                break;
            }
            each_contact_info = each_contact_info ->
next_contact_info;
        }
    }
    return all_contact_info;
}

phonebook * delete_phonebook_by_name(phonebook * all_contact_info, char
* name){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    phonebook * blank = blank_phonebook();
    phonebook * temp1 = blank_phonebook();
    if(strcmp(each_contact_info -> single_contact_info -> name, name)

```

```

== 0){
    blank = each_contact_info -> next_contact_info;
    free(each_contact_info);
    all_contact_info = blank;
}else{
    while(each_contact_info -> next_contact_info != NULL){
        if(strcmp(each_contact_info -> next_contact_info ->
single_contact_info -> name, name) == 0){
            each_contact_info -> next_contact_info =
each_contact_info -> next_contact_info -> next_contact_info;
            break;
        }
        each_contact_info = each_contact_info ->
next_contact_info;
    }
}
return all_contact_info;
}

phonebook * delete_phonebook_by_email(phonebook * all_contact_info, char
* email){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    phonebook * blank = blank_phonebook();
    phonebook * temp1 = blank_phonebook();
    if(strcmp(each_contact_info -> single_contact_info -> email,
email) == 0){
        blank = each_contact_info -> next_contact_info;
        free(each_contact_info);
        all_contact_info = blank;
    }else{
        while(each_contact_info -> next_contact_info != NULL){
            if(strcmp(each_contact_info -> next_contact_info ->
single_contact_info -> email, email) == 0){
                each_contact_info -> next_contact_info =
each_contact_info -> next_contact_info -> next_contact_info;
                break;
            }
            each_contact_info = each_contact_info ->
next_contact_info;
        }
    }
    return all_contact_info;
}

```

```

void search_phonebook_by_email(phonebook * all_contact_info, char *
email){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    printf(
"-----\n"
"      name\t\tlocation\t\temail\t\t\tphone\n"
"-----\n"
"      \n"
    );

    while(each_contact_info != NULL){
        if(strcmp(each_contact_info -> single_contact_info -> email,
email) == 0){
            printf(
                "%.6s\t\t"
                "%.8s\t\t\t"
                "%.14s\t\t"
                "%.12s\n",
                each_contact_info -> single_contact_info -> name,
                each_contact_info -> single_contact_info ->
location,
                each_contact_info -> single_contact_info ->
email,
                each_contact_info -> single_contact_info -> phone
            );
        }
        each_contact_info = each_contact_info -> next_contact_info;
    }
}

void search_phonebook_by_location(phonebook * all_contact_info, char *
location){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    printf(
"-----\n"
"      name\t\tlocation\t\temail\t\t\tphone\n"

```

```

-----\n"
        "\n"
    );

    while(each_contact_info != NULL){
        if(strcmp(each_contact_info -> single_contact_info ->
location, location) == 0){
            printf(
                "%.6s\t\t\t"
                "%.8s\t\t\t\t"
                "%.14s\t\t\t"
                "%.12s\n",
                each_contact_info -> single_contact_info -> name,
                each_contact_info -> single_contact_info ->
location,
                each_contact_info -> single_contact_info ->
email,
                each_contact_info -> single_contact_info -> phone
            );
        }
        each_contact_info = each_contact_info -> next_contact_info;
    }
}

void search_phonebook_by_name(phonebook * all_contact_info, char *
name){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    printf(
"-----\n"
        "name\t\t\tlocation\t\t\temail\t\t\t\tphone\n"
"-----\n"
        "\n"
    );

    while(each_contact_info != NULL){
        if(strcmp(each_contact_info -> single_contact_info -> name,
name) == 0){
            printf(

```

```

        "%.6s\t\t"
        "%.8s\t\t\t"
        "%.14s\t\t"
        "%.12s\n",
        each_contact_info -> single_contact_info -> name,
        each_contact_info -> single_contact_info ->
location,
        each_contact_info -> single_contact_info ->
email,
        each_contact_info -> single_contact_info -> phone
    );
}
each_contact_info = each_contact_info -> next_contact_info;
}
}

void screen_phonebook(phonebook * all_contact_info){
    phonebook * each_contact_info = blank_phonebook();
    each_contact_info = all_contact_info;

    printf(

"-----\n"
"-----\n"
    "name\t\tlocation\t\temail\t\t\tphone\n"

"-----\n"
"-----\n"
    "\n"

    );

    while(each_contact_info != NULL){
        printf(
            "%.6s\t\t"
            "%.8s\t\t\t"
            "%.14s\t\t"
            "%.12s\n",
            each_contact_info -> single_contact_info -> name,
            each_contact_info -> single_contact_info -> location,
            each_contact_info -> single_contact_info -> email,
            each_contact_info -> single_contact_info -> phone
        );
        each_contact_info = each_contact_info -> next_contact_info;
    }
}

```

```

phonebook * insert_phonebook(phonebook * all_contact_info, contact_info
* new_contact_info){
    phonebook * new_contact = blank_phonebook();
    new_contact -> single_contact_info = new_contact_info;

    phonebook * temp = blank_phonebook();
    temp = all_contact_info;

    if(temp == NULL){
        all_contact_info = new_contact;
    }else{
        while(temp -> next_contact_info != NULL){
            temp = temp -> next_contact_info;
        }
        temp -> next_contact_info = new_contact;
    }

    return all_contact_info;
}

contact_info * insert_contact_info(char * name, char * location, char *
email, char * phone){
    contact_info * single_contact = blank_contact_info();
    strcpy(single_contact -> name, name);
    strcpy(single_contact -> location, location);
    strcpy(single_contact -> email, email);
    strcpy(single_contact -> phone, phone);

    return single_contact;
}

contact_info * blank_contact_info(){
    contact_info * blank = (contact_info *)
malloc(sizeof(contact_info));
    return blank;
}

phonebook * blank_phonebook(){
    phonebook * blank = (phonebook *) malloc(sizeof(phonebook));
    blank -> next_contact_info = NULL;
    return blank;
}

phonebook * sample_data(phonebook * all_contact_info){
    contact_info * new_contact_info1 = insert_contact_info("name1",

```

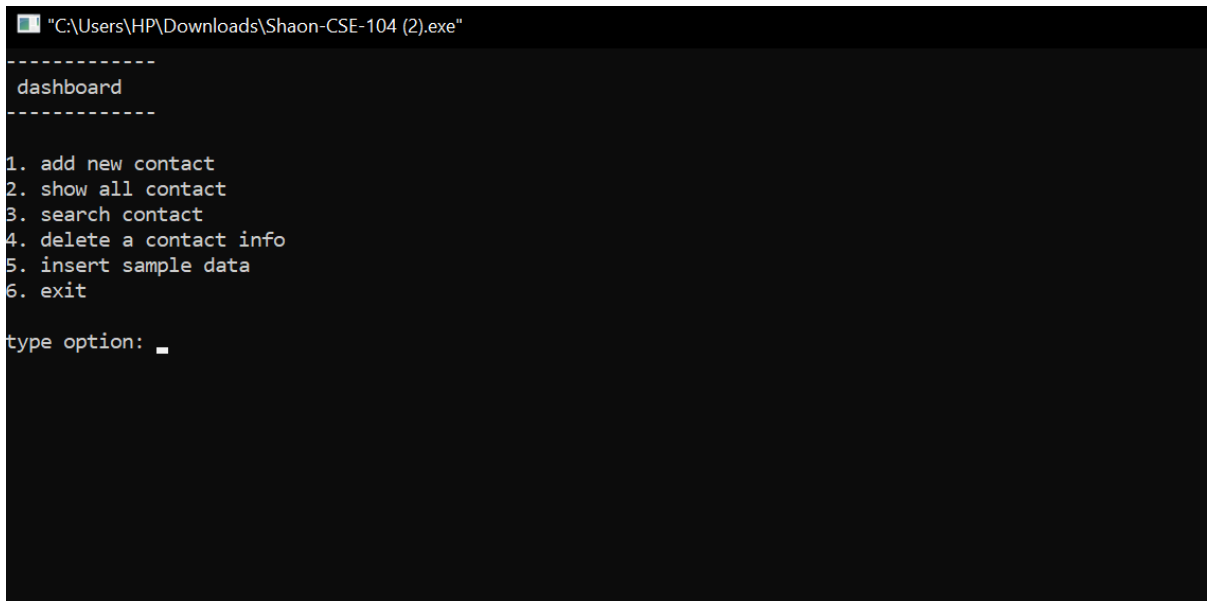


```
"locati1", "mail1@mail.com", "123123123");
    contact_info * new_contact_info2 = insert_contact_info("name2",
"locati2", "mail2@mail.com", "123123123");
    contact_info * new_contact_info3 = insert_contact_info("name3",
"locati3", "mail3@mail.com", "123123123");
    contact_info * new_contact_info4 = insert_contact_info("name4",
"locati4", "mail4@mail.com", "123123123");

    all_contact_info = insert_phonebook(all_contact_info,
new_contact_info1);
    all_contact_info = insert_phonebook(all_contact_info,
new_contact_info2);
    all_contact_info = insert_phonebook(all_contact_info,
new_contact_info3);
    all_contact_info = insert_phonebook(all_contact_info,
new_contact_info4);

    return all_contact_info;
}
```

## Screenshots



```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"

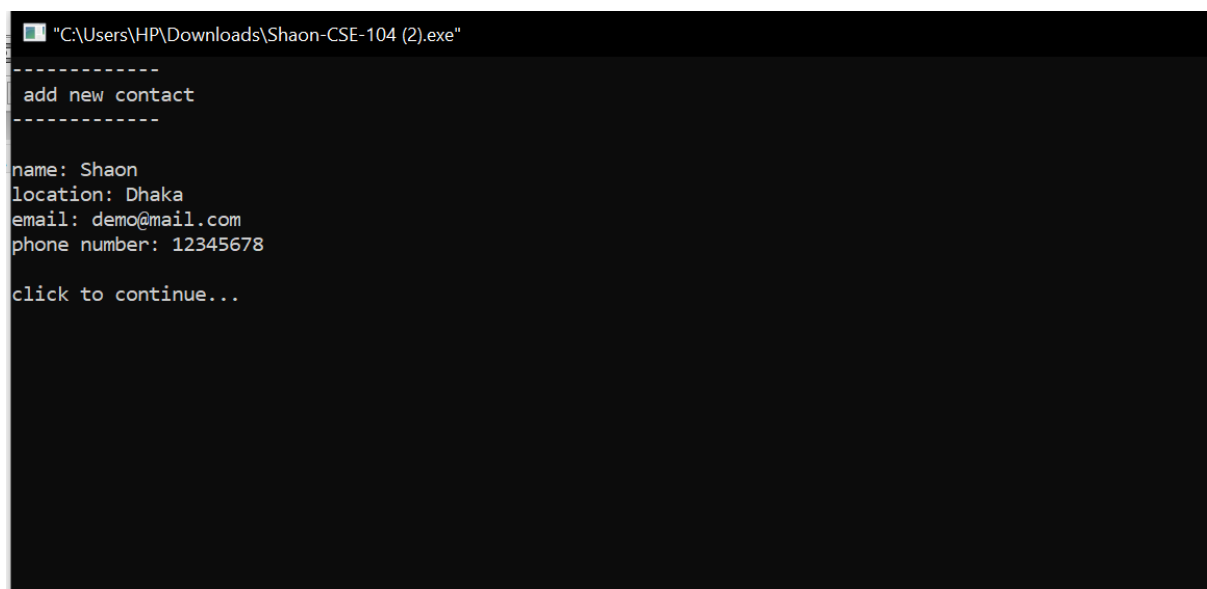
-----
dashboard
-----

1. add new contact
2. show all contact
3. search contact
4. delete a contact info
5. insert sample data
6. exit

type option: _
```

Figure 3: Main Menu interface

When the is executed ,the user will be directed to the main menu interface. The program is introduced with a few lines of texts. Then Six selections are made for the user as the user can choose to add new contact,show all contact,search contact,delete a contact info,insert sample data or exit the program.



```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"

-----
add new contact
-----

name: Shaon
location: Dhaka
email: demo@mail.com
phone number: 12345678

click to continue...
```

Figure 4: add new contact entered (main manu)

The program is asking the user to enter name,location,email, and phone number to add new contact for the user.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
name           location           email           phone
-----
Shaon          Dhaka                demo@mail.com   12345678
click to continue...
```

Figure 5:the contact add successfully.

As shown the interface the contact number is add successfully, after the user entered the show all contacts.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
search contact
-----
1. search by name
2. search by email
type option:
```

Figure 6: search a contact by name or email.

The program is asking the user to enter name or email, to search contact for the user.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
search contact
-----
1. search by name
2. search by email

type option: 1
type your name: Shaon
-----
name            location            email            phone
-----
Shaon           Dhaka              demo@mail.com    12345678
click to continue...
```

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
search contact
-----
1. search by name
2. search by email

type option: 2
type your email: demo@mail.com
-----
name            location            email            phone
-----
Shaon           Dhaka              demo@mail.com    12345678
click to continue...
```

Figure 7: the search contact work done successfully.

As shown the interface the search contact work was done successfully by name and email.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"

-----
delete contact
-----

1. delete by name
2. delete by email

type option: _
```

Figure 8: delete a contact by name or email.

The program is asking the user to enter name or email, to delete a contact for the user.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"

-----
delete contact
-----

1. delete by name
2. delete by email

type option: 1
type your name: Shaon

click to continue...
```

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
name          location          email          phone
-----
click to continue...
_
```

Figure 9: The contact is deleted successfully.

As shown the interface the contact number is deleted successfully, after the user entered the show all contacts.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
dashboard
-----
1. add new contact
2. show all contact
3. search contact
4. delete a contact info
5. insert sample data
6. exit
type option: 5_
```

Figure 10 : Insert sample data.

Here I insert some sample data.

```
"C:\Users\HP\Downloads\Shaon-CSE-104 (2).exe"
-----
name          location          email          phone
-----
name1         locati1           mail1@mail.com 123123123
name2         locati2           mail2@mail.com 123123123
name3         locati3           mail3@mail.com 123123123
name4         locati4           mail4@mail.com 123123123
click to continue...
```

Fig 11 : Insert Sample data Output.

As shown the interface the insert sample data output , after the user entered the show all contacts.

# Chapter 3

## Conclusion

### Learning Outcomes

The Program makes the user simpler to connect to his contact. The contact personal information is stored under a single number this would benefit the user to easily search and locate his required contact. This program deals with six operations add new contact, show all contact, search contact, delete a contact info, insert sample data and exit according to the user's choice.

### Future Scopes

- Contact exchange solutions for easy and quick access to sharing business contacts.
- Real-time messaging and chat options, facilitating productive interactions.



## References

- (1) ICQ Home Page. <http://www.icq.com/>. July
- (2) 2001. 2. Jacknis, M., Sawhney, N., Schmandt, C. GarblePhone: auditory lurking. July 2000.
- (3) Liechti O, Siefer N and Ichikawa T. A Non-obtrusive User Interface for Increasing Social Awareness on the World Wide Web. Personal Technologies 3 (1&2), 1999: 22-32.
- (4) Schmidt, A., Takaluoma, A. and Mäntyjärvi, J.. Context-Aware Telephony over WAP. Personal Technologies Volume 4(4), September 2000, pp225-229

**THANKS**