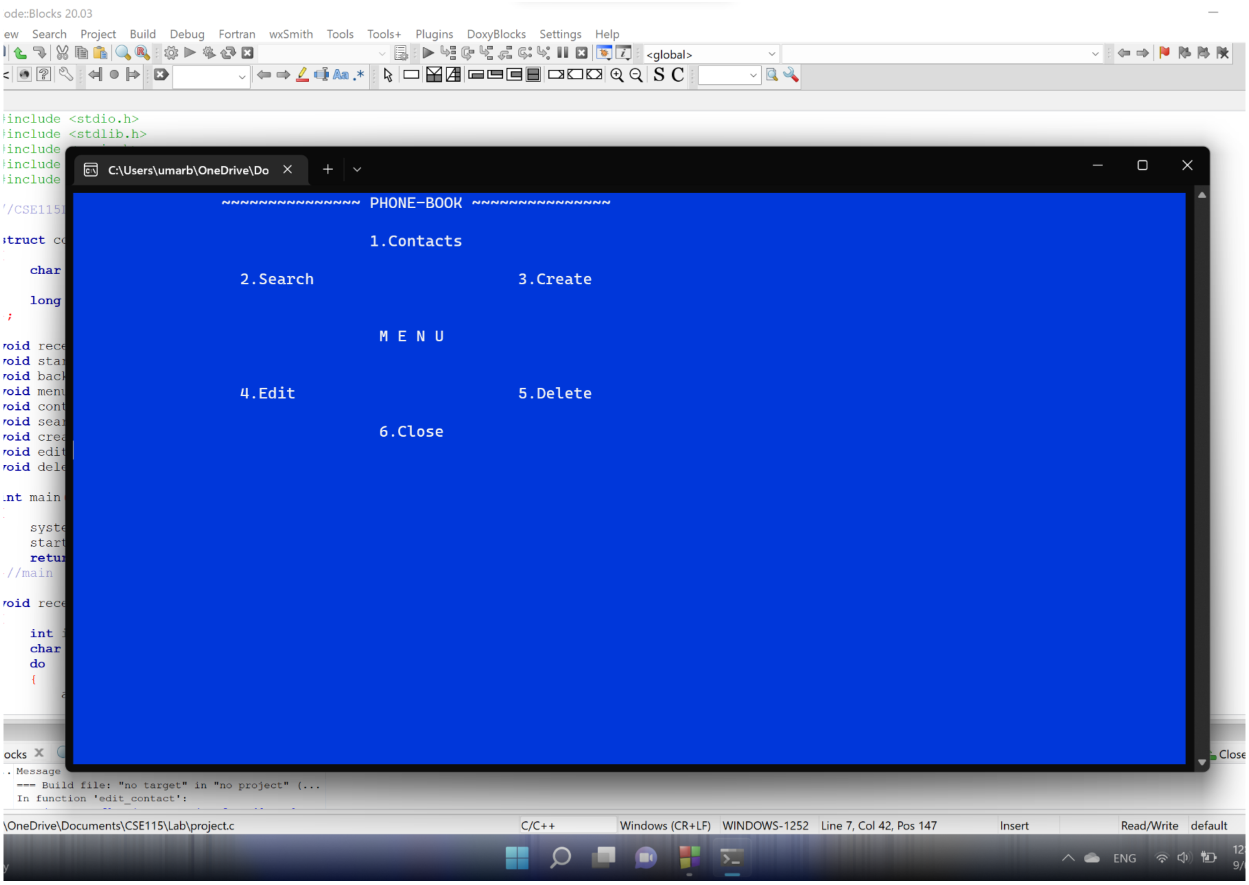
# CSE115L.4: Project

*Phone-book Application*



ode::Blocks 20.03
ew Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
C:\Users\umarb\OneDrive\Do
ocks
Close
\OneDrive\Documents\CSE115\Lab\project.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 7, Col 42, Pos 147
Insert
Read/Write
default
ENG

```
#include <stdio.h>
#include <stdlib.h>
#include
#include
#include

//CSE115

struct co

    char

    long
};

void rece
void star
void back
void menu
void cont
void sear
void crea
void edit
void dele

int main

    syste
    start
    retur
//main

void rece

    int i
    char
    do
    {
```

```
~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                1.Contacts

   2.Search                     3.Create


                 M E N U


   4.Edit                       5.Delete

                 6.Close
```

```
. Message
=== Build file: "no target" in "no project" (...
In function 'edit_contact':
```

## Presented to you by the members of the G1 group

| Group Name | Group Member | Student ID | Project Idea |
|---|---|---|---|
| G1 | Umar Hasan | 2222739642 | Phone-book application |
| | Midhat Bin Shazzad | 2222560642 | |
| | Shafin Bin Imran | 2221541642 | |
| | Azowad Islam | 2221581642 | |
| | Md. Abu Sayem | 2221657642 | |

Date of Submission: 07.09.2022

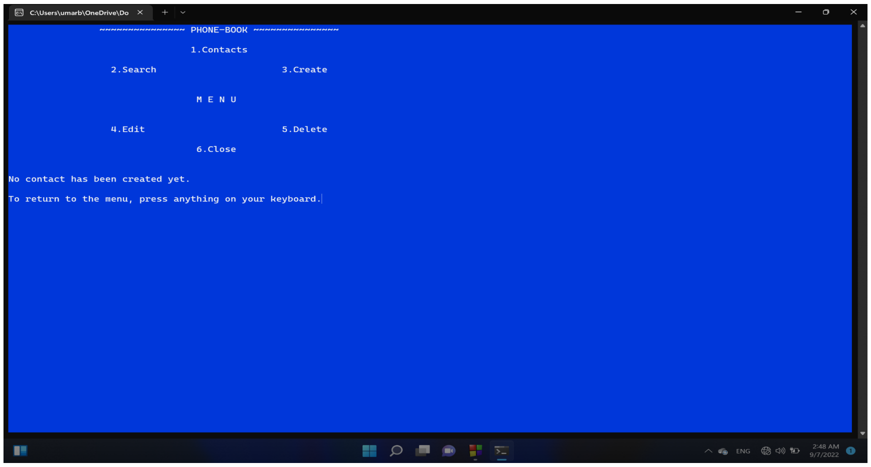
## ABOUT

This project’s application is similar to the contact management system. But, the main difference is that you can add some additional information like, gender, email, and address. The source code of the application has been written in C.

## FEATURES

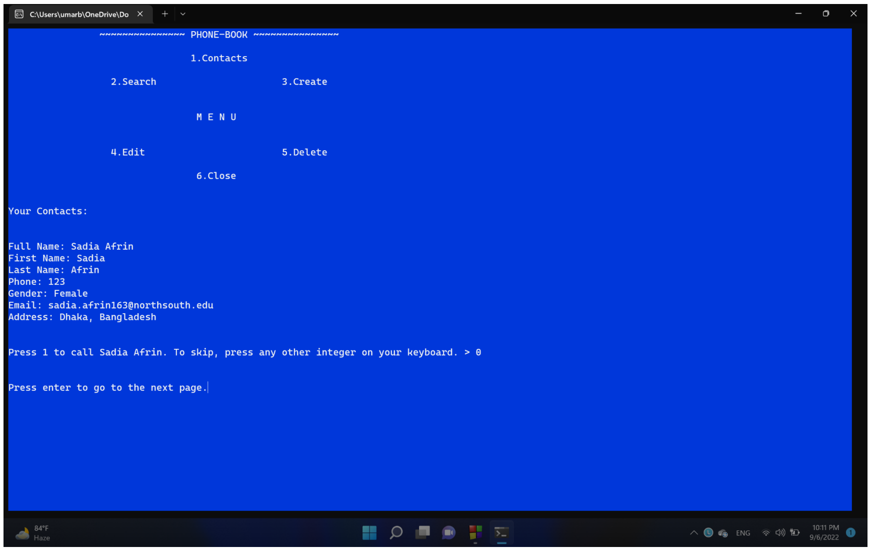
The application contains the following six features:

1. Contacts: This feature lets the users check their contacts list and call anyone from it.
   Some sample screenshots are attached below.
   Before creating any contact:

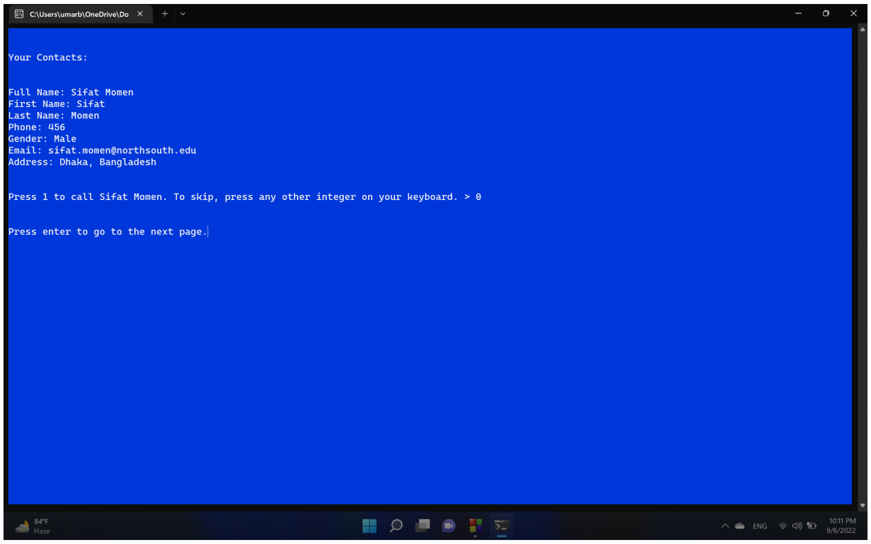


```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~
                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close


No contact has been created yet.

To return to the menu, press anything on your keyboard.
```

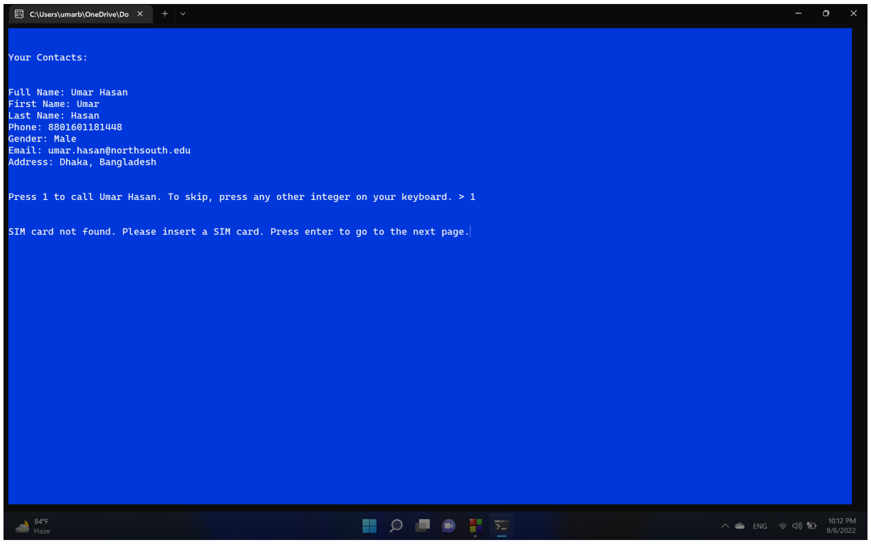
   After creating three contacts:



```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close


Your Contacts:


Full Name: Sadia Afrin
First Name: Sadia
Last Name: Afrin
Phone: 123
Gender: Female
Email: sadia.afrin163@northsouth.edu
Address: Dhaka, Bangladesh


Press 1 to call Sadia Afrin. To skip, press any other integer on your keyboard. > 0


Press enter to go to the next page.
```

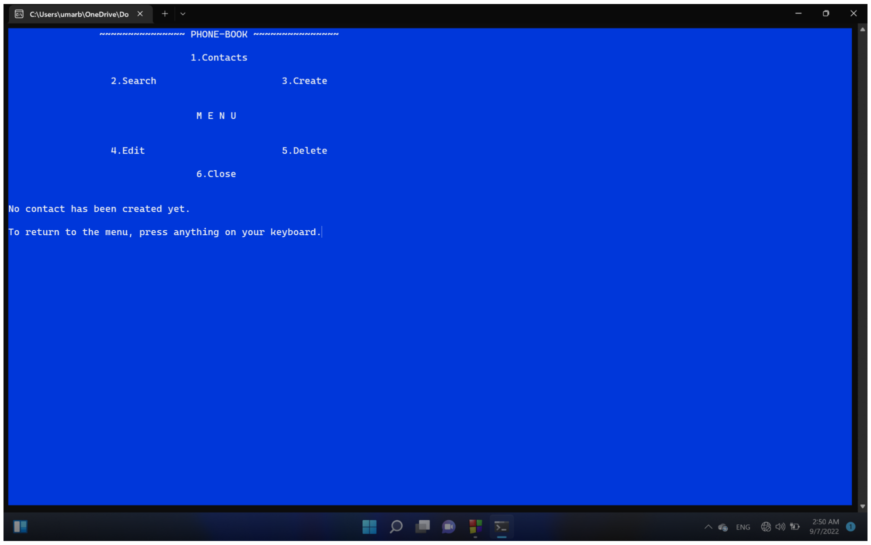


```


Your Contacts:


Full Name: Sifat Momen
First Name: Sifat
Last Name: Momen
Phone: 456
Gender: Male
Email: sifat.momen@northsouth.edu
Address: Dhaka, Bangladesh


Press 1 to call Sifat Momen. To skip, press any other integer on your keyboard. > 0


Press enter to go to the next page.
```



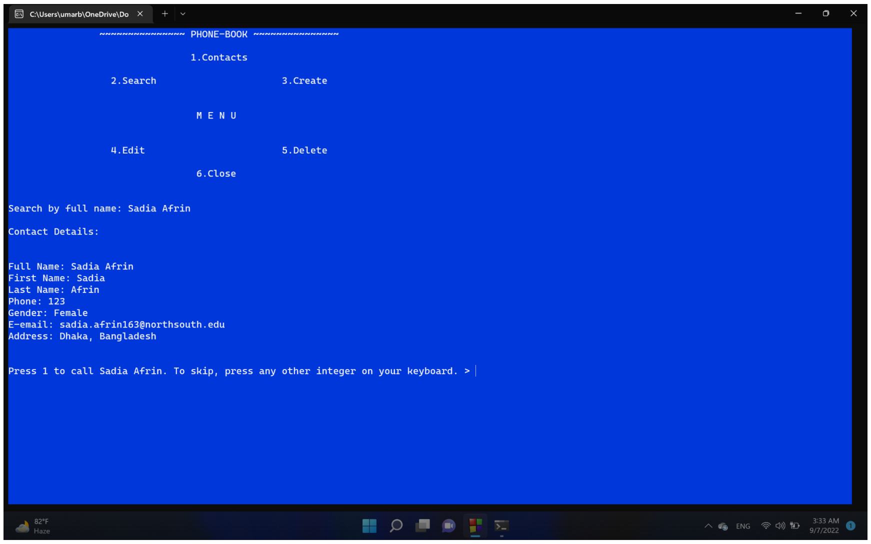
```


Your Contacts:


Full Name: Umar Hasan
First Name: Umar
Last Name: Hasan
Phone: 8801601181448
Gender: Male
Email: umar.hasan@northsouth.edu
Address: Dhaka, Bangladesh


Press 1 to call Umar Hasan. To skip, press any other integer on your keyboard. > 1


SIM card not found. Please insert a SIM card. Press enter to go to the next page.
```

2. Search: This feature lets the users search and call anyone from their contacts list. Some sample screenshots are attached below.
   Before creating any contact:



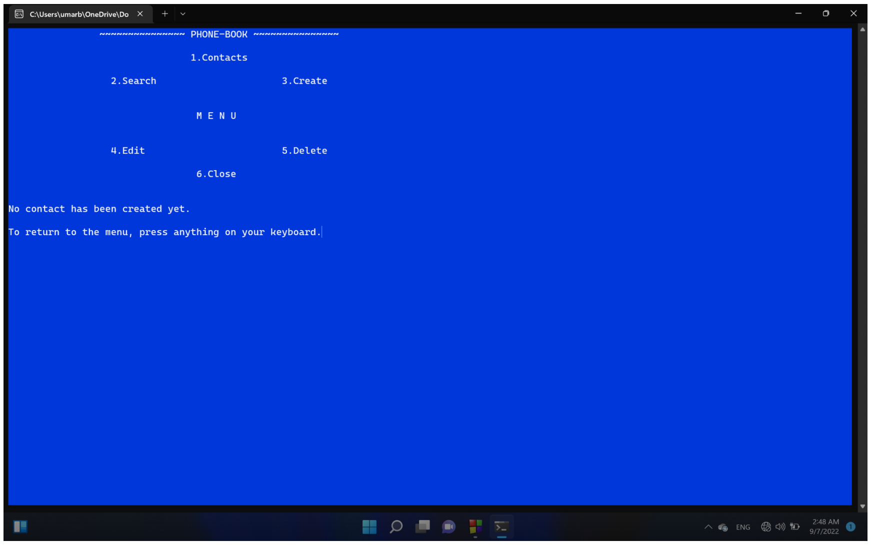
```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close


No contact has been created yet.

To return to the menu, press anything on your keyboard.
```

After creating contacts:



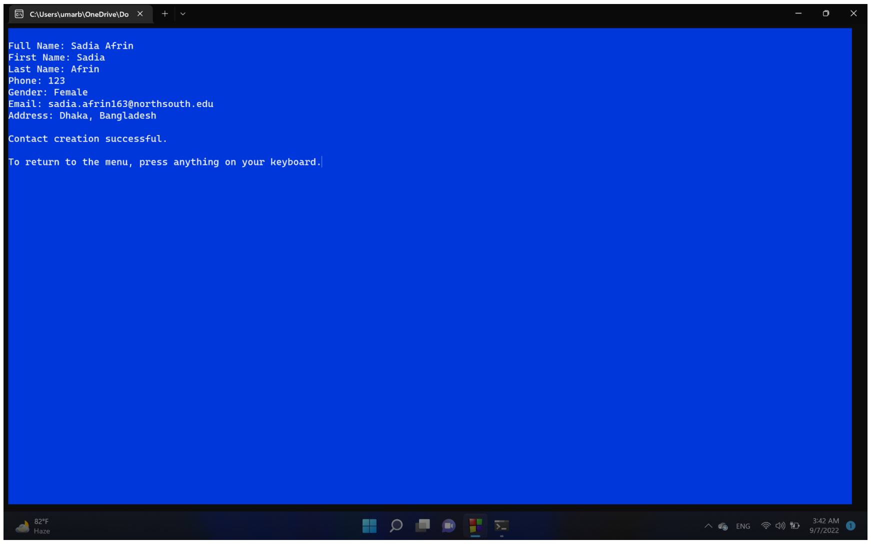
```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                        3.Create


                                 M E N U


                  4.Edit                          5.Delete

                                 6.Close


Search by full name: Sadia Afrin

Contact Details:


Full Name: Sadia Afrin
First Name: Sadia
Last Name: Afrin
Phone: 123
Gender: Female
E-email: sadia.afrin163@northsouth.edu
Address: Dhaka, Bangladesh


Press 1 to call Sadia Afrin. To skip, press any other integer on your keyboard. >
```

3. Create: This feature lets the users create contacts and save them in their contacts list. Some sample screenshots are attached below.
Before creating any contact:



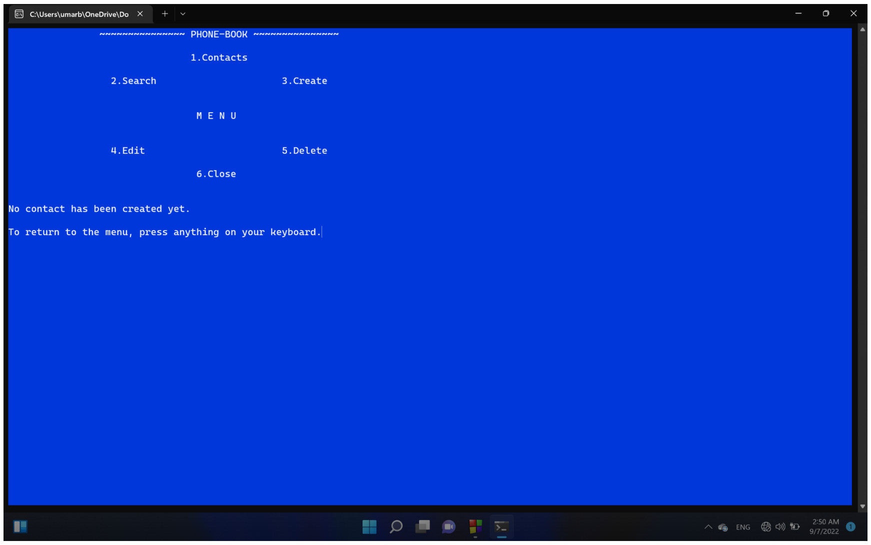
```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                        3.Create


                                 M E N U


                  4.Edit                          5.Delete

                                 6.Close


No contact has been created yet.

To return to the menu, press anything on your keyboard.
```

After creating contacts:



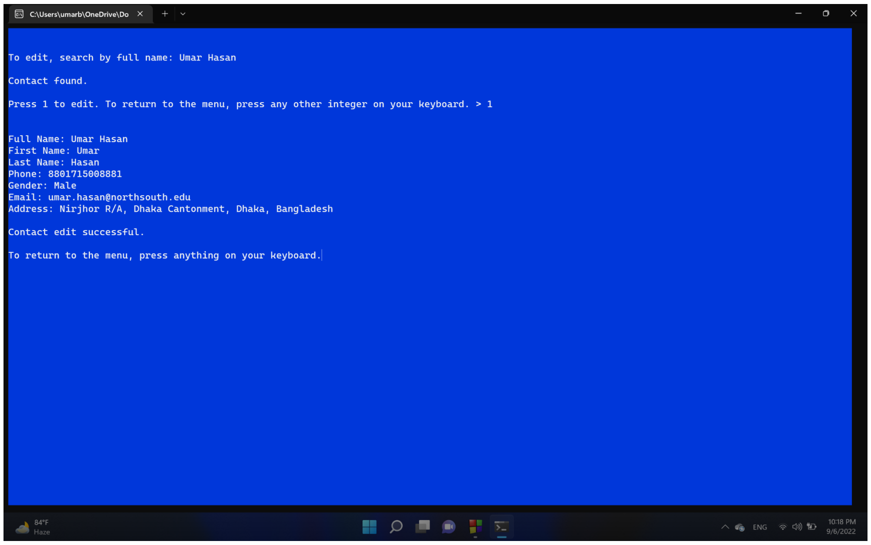
```
Full Name: Sadia Afrin
First Name: Sadia
Last Name: Afrin
Phone: 123
Gender: Female
Email: sadia.afrin163@northsouth.edu
Address: Dhaka, Bangladesh

Contact creation successful.

To return to the menu, press anything on your keyboard.
```

4. Edit: This feature lets the users edit contacts and save the edited version in their contacts list. Some sample screenshots are attached below.
   Before creating any contact:



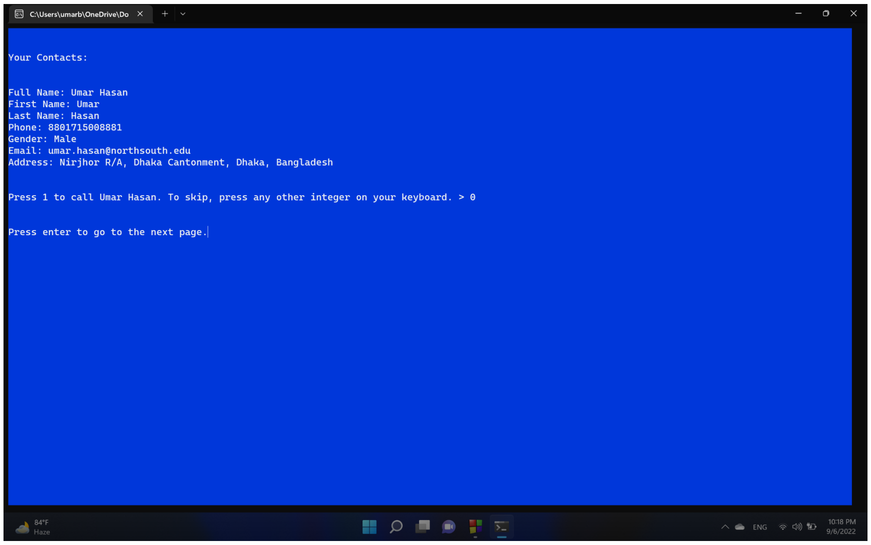
```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close


No contact has been created yet.

To return to the menu, press anything on your keyboard.
```

After creating contacts:



```
To edit, search by full name: Umar Hasan

Contact found.

Press 1 to edit. To return to the menu, press any other integer on your keyboard. > 1


Full Name: Umar Hasan
First Name: Umar
Last Name: Hasan
Phone: 8801715008881
Gender: Male
Email: umar.hasan@northsouth.edu
Address: Nirjhor R/A, Dhaka Cantonment, Dhaka, Bangladesh

Contact edit successful.

To return to the menu, press anything on your keyboard.
```

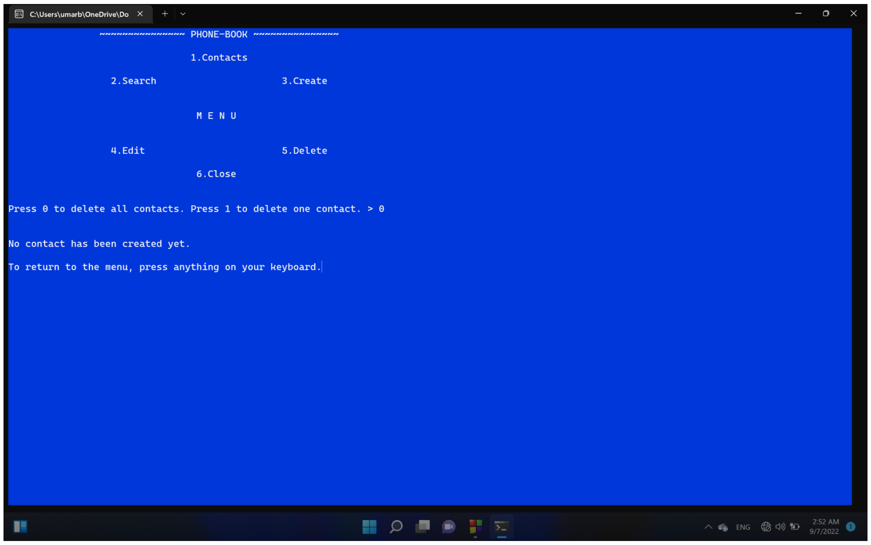
The contact list is updated after editing:



```
Your Contacts:


Full Name: Umar Hasan
First Name: Umar
Last Name: Hasan
Phone: 8801715008881
Gender: Male
Email: umar.hasan@northsouth.edu
Address: Nirjhor R/A, Dhaka Cantonment, Dhaka, Bangladesh


Press 1 to call Umar Hasan. To skip, press any other integer on your keyboard. > 0


Press enter to go to the next page.
```

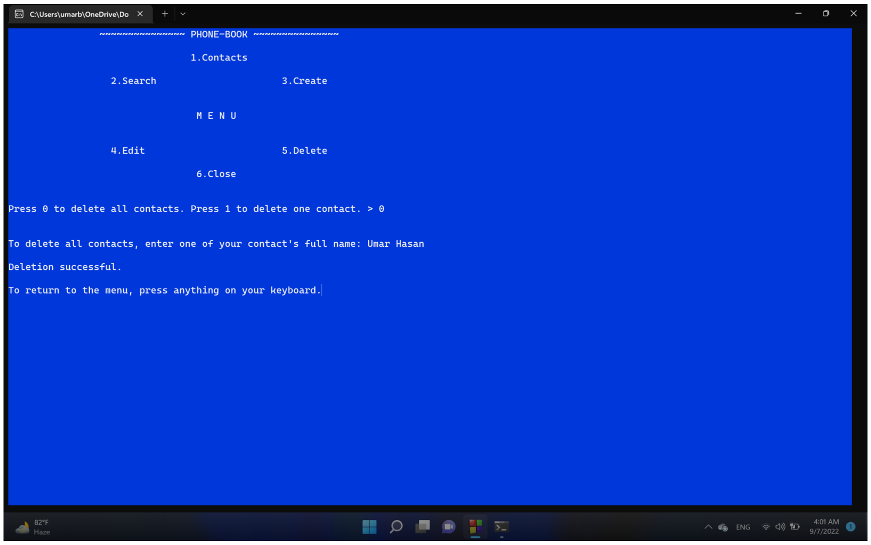
5. Delete: This feature lets the users delete contacts from their contacts list. Some sample screenshots are attached below.
   When deleting all contacts:

   Before creating any contact:



```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close


Press 0 to delete all contacts. Press 1 to delete one contact. > 0


No contact has been created yet.

To return to the menu, press anything on your keyboard.
```

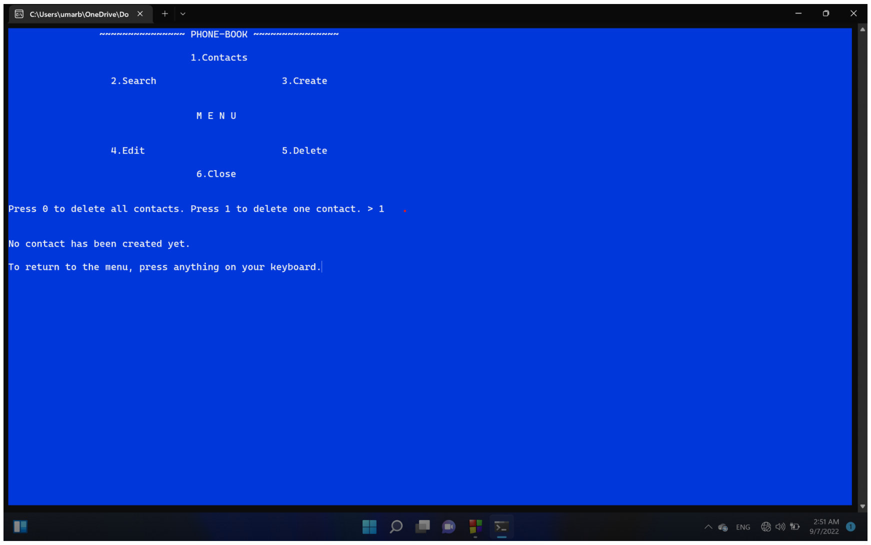
   After creating contacts:



```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close


Press 0 to delete all contacts. Press 1 to delete one contact. > 0


To delete all contacts, enter one of your contact's full name: Umar Hasan

Deletion successful.

To return to the menu, press anything on your keyboard.
```

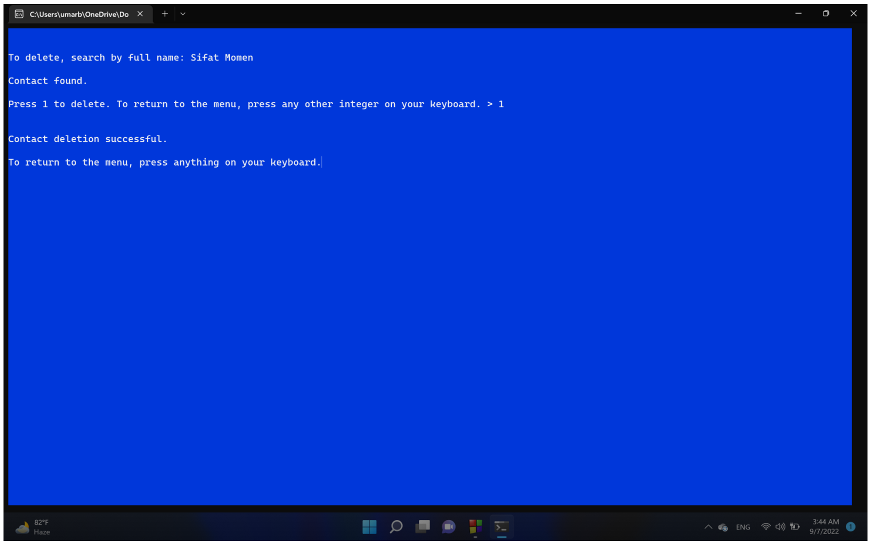
When deleting one contact:

Before creating any contact:



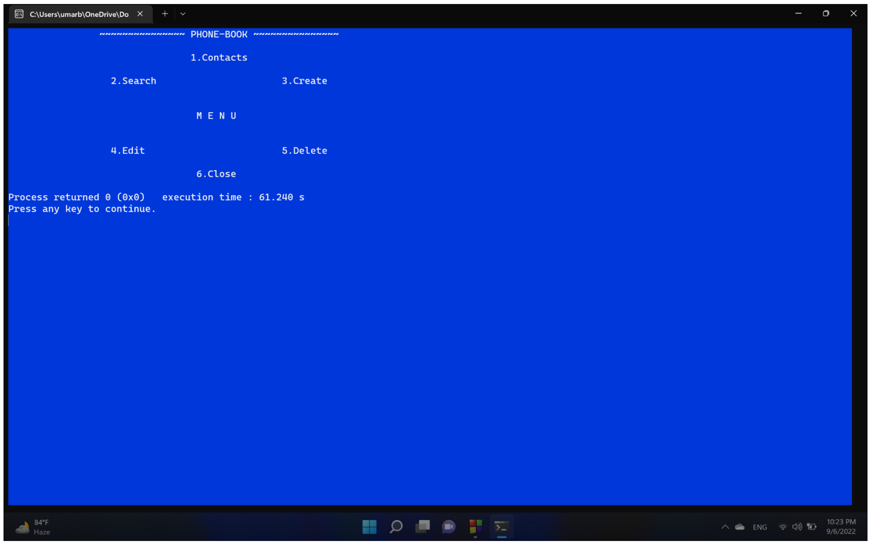
```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                        3.Create

                                 M E N U

                  4.Edit                          5.Delete

                                 6.Close

Press 0 to delete all contacts. Press 1 to delete one contact. > 1


No contact has been created yet.

To return to the menu, press anything on your keyboard.
```

After creating contacts:



```
To delete, search by full name: Sifat Momen

Contact found.

Press 1 to delete. To return to the menu, press any other integer on your keyboard. > 1


Contact deletion successful.

To return to the menu, press anything on your keyboard.
```

6. Close: This feature lets the users close the phone-book application and exit from it. A sample screenshot is attached below.



```
                ~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~

                                1.Contacts

                  2.Search                      3.Create


                                 M E N U


                  4.Edit                        5.Delete

                                 6.Close

Process returned 0 (0x0)   execution time : 61.240 s
Press any key to continue.
```

These are the six core features of our phone-book application. To check the source code of the application, please go to the next page. Thank you.

## SOURCE CODE

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

//CSE115L Project: Phone-book application using C

struct contact
{
   char full_name[100], first_name[50], last_name[50], gender[7], email[100], address[100];

   long long int phone;
};

void receive();
void start();
void menu();
void contacts_contact();
void search_contact();
void create_contact();
void edit_contact();
void delete_contact();

int main()
{
   system("color 1f"); //console background: blue
   start();
   return 0;
}//main

//Reads and stores the user character inputs.
void receive(char *full_name)
{
   int i = 0, j;
   char a, b;
   do
   {
      a = getch();

      if(a != 8 && a != 13)
      {
            *(full_name + i) = a;
            putch(a);
            i++;
        }

        if(a == 8)
        {
            if(i > 0)
            {
                i--;
            }

        system("cls");

            for(j = 0; j < i; j++)
            {
                b = *(full_name + j);
                putch(b);
            }
        }//if(a == 8)
    }while(a != 13);

    *(full_name + i) = '\0';
}//receive

//Opens the menu screen of the application.
void start()
{
    menu();
}

//Runs the menu screen of the application.
void menu()
{
    system("cls");

    printf("\t\t~~~~~~~~~~~~~~~ PHONE-BOOK ~~~~~~~~~~~~~~~");

    printf("\n\n\t\t\t\t1.Contacts \n\n\t\t  2.Search \t\t\t3.Create \n\n\n\t\t\t\t M E N U\n \n\n\t\t  4.Edit
\t\t\t5.Delete \n\n\t\t\t\t 6.Close\n");


    switch(getch())
    {
```

```
        case '1':
            contacts_contact();
            break;
        case '2':
            search_contact();
            break;
        case '3':
            create_contact();
            break;
        case '4':
            edit_contact();
            break;
        case '5':
            delete_contact();
            break;
        case '6':
            exit(0);
            break;
        default:

        system("cls");

        printf("To select from the menu, press anything from 1 to 6 only.");

        printf("\n\nTo return to the menu, press anything on your keyboard.");

        getch();
        menu();
    }//switch
}//menu

//Prints out all the user inputs stored as contact information.
void contacts_contact()
{
    FILE *primary_file;

    int call, count = 0;

    struct contact original_contact;


    primary_file = fopen("project", "rb");

    if(primary_file == NULL)
    {
        printf("\n\nNo contact has been created yet.");

        printf("\n\nTo return to the menu, press anything on your keyboard.");

        getch();
        system("cls");
        menu();
    }

    while(fread(&original_contact, sizeof(original_contact), 1, primary_file) == 1)
    {
        if(strcmp(original_contact.full_name, "Deleted Contact") != 0)
        {
            printf("\n\nYour Contacts:\n\n");

            printf("\nFull Name: %s\nFirst Name: %s\nLast Name: %s\nPhone: %lld\nGender:
%s\nEmail: %s\nAddress: %s\n",
                    original_contact.full_name, original_contact.first_name,original_contact. last_name,
original_contact.phone, original_contact.gender, original_contact.email,
original_contact.address);

            printf("\n\nPress 1 to call %s. To skip, press any other integer on your keyboard. > ",
original_contact.full_name);
            scanf("%d", &call);

            if(call == 1)
            {
                printf("\n\nSIM card not found. Please insert a SIM card. Press enter to go to the next
page.");
            }

            else
                printf("\n\nPress enter to go to the next page.");

            count++;
        }

        else
            continue;

        getch();
        system("cls");
    }//while
    if(count == 0)
        printf("All of the created contacts were deleted one by one.");

    fclose(primary_file);

    printf("\n\nTo return to the menu, press anything on your keyboard.");

    getch();
    system("cls");
    menu();
}//contacts

/*
    Reads a user character input as a contact’s full name, searches for the identical data stored
as a contact’s full name in the memory,
    and then prints the rest of the information of that specific contact if the input data matches
with the data stored in the memory.
 */
void search_contact()
{
    char name[100];

    FILE *primary_file;

    int call;

    struct contact original_contact;


    primary_file = fopen("project", "rb");

    if(primary_file == NULL)
    {
        printf("\n\nNo contact has been created yet.");

        printf("\n\nTo return to the menu, press anything on your keyboard.");

        getch();
        system("cls");
        menu();
    }

    else
    {
        printf("\n\nSearch by full name: ");
        receive(name);

        while(fread(&original_contact, sizeof(original_contact), 1, primary_file) == 1)
        {
            if(strcmp(original_contact.full_name, name) == 0 && strcmp(original_contact.full_name,
"Deleted Contact") != 0)
            {
                printf("\n\nContact Details:\n\n");

                printf("\nFull Name: %s\nFirst Name: %s\nLast Name: %s\nPhone: %lld\nGender:
%s\nE-email: %s\nAddress: %s\n",
                        original_contact.full_name, original_contact.first_name,
original_contact.last_name, original_contact.phone, original_contact.gender,
original_contact.email, original_contact.address);

                printf("\n\nPress 1 to call %s. To skip, press any other integer on your keyboard. > ",
original_contact.full_name);
                scanf("%d", &call);

                if(call == 1)
                {
                    printf("\n\nSIM card not found. Please insert a SIM card.");
                }

                break;
            }//if

            else
            {
                printf("\n\nNothing found.");

                break;
            }
        }//while

        fclose(primary_file);

        printf("\n\nTo return to the menu, press anything on your keyboard.");

        getch();
        system("cls");
        menu();
    }//else
}

//Reads and stores the user inputs as contact information.
void create_contact()
{
    system("cls");

    FILE *primary_file;

    struct contact original_contact;


    primary_file = fopen("project", "ab+");

    printf("\nFull Name: ");
    receive(original_contact.full_name);

    printf("\nFirst Name: ");
    receive(original_contact.first_name);

    printf("\nLast Name: ");
    receive(original_contact.last_name);

    printf("\nPhone: ");
    scanf("%lld", &original_contact.phone);

    printf("Gender: ");
    receive(original_contact.gender);

    printf("\nEmail: ");
    receive(original_contact.email);

    printf("\nAddress: ");
    receive(original_contact.address);

    fwrite(&original_contact, sizeof(original_contact), 1, primary_file);

    fflush(stdin);

    printf("\n\nContact creation successful.");

    fclose(primary_file);
    printf("\n\nTo return to the menu, press anything on your keyboard.");

    getch();
    system("cls");
    menu();
}//create

//Reads new user inputs and replaces them with the old user inputs stored in the memory.
void edit_contact()
{
    char name[100];

    FILE *primary_file;

    int edit, flag = 0;

    struct contact original_contact, edited_contact;


    primary_file = fopen("project", "rb+");

    if(primary_file == NULL)
    {
        printf("\n\nNo contact has been created yet.");

        printf("\n\nTo return to the menu, press anything on your keyboard.");

        getch();
        system("cls");
        menu();
    }

    else
    {
        system("cls");

        printf("\n\nTo edit, search by full name: ");
        receive(name);

        while(fread(&original_contact, sizeof(original_contact), 1, primary_file) == 1)
        {
            if(strcmp(name, original_contact.full_name) == 0)
            {
                printf("\n\nContact found.");
                printf("\n\nPress 1 to edit. To return to the menu, press any other integer on your keyboard. > ");
                scanf("%d", &edit);

                if(edit != 1)
                    break;

                else if(edit == 1)
                {
                    printf("\n\nFull Name: ");
                    receive(edited_contact.full_name);

                    printf("\nFirst Name: ");
                    receive(edited_contact.first_name);

                    printf("\nLast Name: ");
                    receive(edited_contact.last_name);

                    printf("\nPhone: ");
                    scanf("%lld",&edited_contact.phone);

                    printf("Gender: ");
                    receive(edited_contact.gender);

                    printf("\nEmail: ");
                    receive(edited_contact.email);

                    printf("\nAddress: ");
                    receive(edited_contact.address);

                    fseek(primary_file, -sizeof(original_contact), SEEK_CUR);

                    fwrite(&edited_contact, sizeof(original_contact), 1, primary_file);

                    flag = 1;

                    break;
                }
            }//if

            fflush(stdin);
        }//while
        if(flag == 1)
        {
            printf("\n\nContact edit successful.");
        }

        else
        {
            printf("\n\nNothing found.");
        }

        fclose(primary_file);
    }//else

    printf("\n\nTo return to the menu, press anything on your keyboard.");

    getch();
    system("cls");
    menu();
}//edit

/*
    Reads a user character input as a contact’s full name and searches for the identical data
stored as a contact’s full name in the memory.
    And if the input data matches with the data stored in the memory, based on another user
input that it reads at first,
    it either deletes only the information of that specific contact or deletes every contact
information in the application.
 */
void delete_contact()
{
    int deleted;

    printf("\n\nPress 0 to delete all contacts. Press 1 to delete one contact. > ");
    scanf("%d", &deleted);

    if(deleted == 1)
    {
        char name[100];

        FILE *primary_file;

        int one_delete, flag = 0;

        struct contact original_contact, edited_contact;
        primary_file = fopen("project", "rb+");

        if(primary_file == NULL)
        {
            printf("\n\nNo contact has been created yet.");

            printf("\n\nTo return to the menu, press anything on your keyboard.");

            getch();
            system("cls");
            menu();
        }

        else
        {
            system("cls");

            printf("\n\nTo delete, search by full name: ");
            receive(name);

            while(fread(&original_contact, sizeof(original_contact), 1, primary_file) == 1)
            {
                if(strcmp(name, "Deleted Contact") == 0)
                    break;
                else if(strcmp(name, original_contact.full_name) == 0)
                {
                    printf("\n\nContact found.");

                    printf("\n\nPress 1 to delete. To return to the menu, press any other integer on your keyboard. > ");
                    scanf("%d", &one_delete);

                    if(one_delete != 1)
                        break;

                    else if(one_delete == 1)
                    {
                        strcpy(edited_contact.full_name, "Deleted Contact");

                        strcpy(edited_contact.first_name, "");

                        strcpy(edited_contact.last_name, "");
                        edited_contact.phone = 00000000000;

                        strcpy(edited_contact.gender, "");

                        strcpy(edited_contact.email, "");

                        strcpy(edited_contact.address, "");

                        //printf("\n\nFull Name: %s %d", edited_contact.full_name, count);

                        fseek(primary_file, -sizeof(original_contact), SEEK_CUR);

                        fwrite(&edited_contact, sizeof(original_contact), 1, primary_file);

                        flag = 1;

                        break;
                    }
                }//else if

                fflush(stdin);
            }//while

            if(flag == 1)
            {
                printf("\n\nContact deletion successful.");
            }

            else
            {
                printf("\n\nNothing found.");
            }

            fclose(primary_file);
        }//else

        printf("\n\nTo return to the menu, press anything on your keyboard.");

        getch();
        system("cls");
        menu();
    }
```

```
else if(deleted == 0)
{
    char name[100];

    FILE *primary_file, *temporary_file;

    int all_delete, flag;

    struct contact original_contact;


    primary_file = fopen("project", "rb");

    if(primary_file == NULL)
    {
        printf("\n\nNo contact has been created yet.");

        printf("\n\nTo return to the menu, press anything on your keyboard.");
    }

    else
    {
        temporary_file = fopen("temp", "wb+");

        if(temporary_file == NULL)
        {
            printf("\n\nOops! Corrupt file detected.");

            printf("\n\nTo return to the menu, press anything on your keyboard.");
        }

        else
        {
            printf("\n\nTo delete all contacts, enter one of your contact's full name: ");
            receive(name);

            fflush(stdin);

            while(fread(&original_contact, sizeof(original_contact), 1, primary_file) == 1)
            {
                if(strcmp(original_contact.full_name, name)!= 0)
                    fwrite(&original_contact, sizeof(original_contact), 1, temporary_file);

                if(strcmp(original_contact.full_name, name) == 0)
```

```
                        flag = 1;
                }

                fclose(primary_file);

                fclose(temporary_file);

                if(flag != 1)
                {
                    printf("\n\nError: That's not one of your contact's full name.");

                    remove("temp.txt");

                    printf("\n\nTo return to the menu, press anything on your keyboard.");
                }

                else
                {
                    remove("project");

                    rename("temp.txt","project");

                    printf("\n\nDeletion successful.");

                    printf("\n\nTo return to the menu, press anything on your keyboard.");
                }
            }//else
        }//else
    }//else if

    getch();
    system("cls");
    menu();
}//delete
```