MINI PROJECT

A mini project is done based on phone book application.

Made by :

RA2011047010074-MEKALA ZETHINDRA .

RA2011047010086-MATHI THARUN.

RA2011047010089-Siddharth Ghosh.

RA2011047010091-DOMA PAVAN SRINIVAS.

PHONE BOOK APPLICATION

AIM : To develop a phone book application that deals with the persons contact details where it performs activities like 1.Adding contacts 2.Listing contacts 3.Modifying contact 4.seraching contact 5.Deleting contact which are the basic functions and also make up the main menu of the phone book application.

OVERVIEW

1.Abstartct

2.Algorithm

3.Flow chart

4.Psuedocode

5.   code and implementation

ABSTRACT

This **mini project in C Phonebook** allows you to perform simple Phonebook operations like in your mobile. You can add, list, modify, search and delete Phonebook-related records. File handling and data structure concepts has been extensively used for almost all functions in this mini project.

Phonebook in C is a console application without graphics.

Phonebook is a very simple mini project in C that can help you understand the basic concepts of functions, file handling and data structure.

Adding new records, listing them, modifying them and updating, search for contacts saved, and deleting the phonebook records are the basic functions which make up the main menu of this Phonebook application (as shown in the main menu screenshot below).

Personal information such as name, sex, father's name, phone number, citizenship number, email and address are asked while adding a record into the Phonebook. These records can then be modified, listed, searched for and removed.

We have used many functions in this mini project. These functions are easy to understand as their name only signifies their respective operations.

- void menu() – This function is used to display the main menu.
- void start() – This functions calls the menu function mentioned above.
- void back() – This function is used to go back to start.
- void addrecord() – It adds a new Phonebook record.
- void listrecord() – This function is used to view list of added records in file.
- void modifyrecord() – This function is used to modify added records.
- void deleterecord() – It deletes record from file.
- void searchrecord() – It searches for added record by name.

ALGORITHM

Step 1 - Start

Step 2 - Declare variables name, age, gender, mobile number, father's name and email-id.

Step 3 - Display "WELCOME TO PHONEBOOK" as the heading and display menu options.

1. ADD CONTACT

2. LIST CONTACT

3. MODIFIY CONTACT

4. SEARCH CONTACT

5. DELETE CONTACT

6. EXIT

Step 4 - Read input data in variables name, age, gender, mobile number, father's name and email-id.

Step 5-Take a number "n" as input and check whether it is between 1 and 6 or equal to 1 and 6

Step 6 - For n=1 (add new), display "enter name, enter age, enter gender, enter mobile number, enter father's name and enter email-id". Then read the input data given to these variables and store it in system.
Then display "contact saved successfully"

Return to menu after finishing operation.

Step 7 - For n=2 (list), print "no contact added", if there is no data stored in system. Otherwise, print all data stored in the system.

Return to menu after finishing operation.

Step 8 - For n= 3 (modify), display "enter contact name to be modified" and read the input.

If input data matches with data stored in name variable, display "enter name, enter age, enter gender, enter mobile number, enter father's name and enter email-id" and read the new input given to the variables, modify the data and store it in the system. Then display "contact modified successfully" Return to menu after finishing operation.

Otherwise, if input data does not match with data stored in name variable, print " no contact to modify." Return to menu after finishing operation.

Step 9 - For n=4 (search), display "enter the contact name" and read the input.

If the input data matches with data stored in name variable, print all data stored under it. Return to menu after finishing operation.

Otherwise, if input data does not match with data stored in name variable, print "contact doesn't exist." Return to menu after finishing operation.
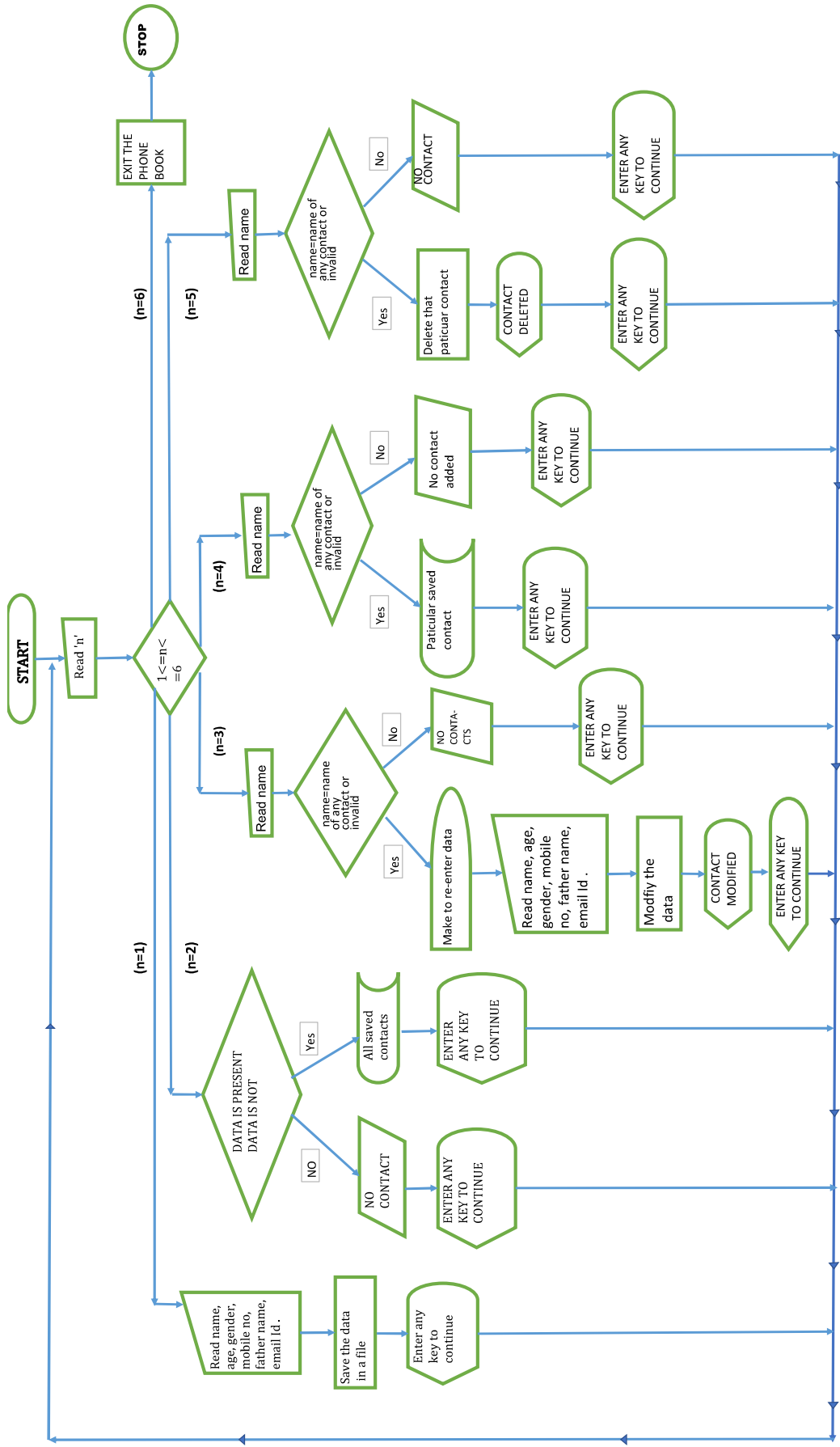
Step 10 - For n=5 (delete), display "enter the contact name to be deleted" and read the input

If the input data matches with data stored in name variable, then delete all the data stored under that name. Then display "contact deleted successfully" Return to menu after finishing operation.

Otherwise if input data does not match with data stored in name variable, print "no contact to delete." Return to menu after finishing operation.

Step 11 - For n=6 (exit), exit from phone book application

Step 12 - Stop

START

Read 'n'

$1 <= n <= 6$

(n=1)

Read name, age, gender, mobile no, father name, email Id.

Save the data in a file

Enter any key to continue

(n=2)

DATA IS PRESENT DATA IS NOT

Yes → All saved contacts → ENTER ANY KEY TO CONTINUE

NO → NO CONTACT → ENTER ANY KEY TO CONTINUE

(n=3)

Read name

name=name of any contact or invalid

Yes → Make to re-enter data → Read name, age, gender, mobile no, father name, email Id. → Modify the data → CONTACT MODIFIED → ENTER ANY KEY TO CONTINUE

No → NO CONTA-CTS → ENTER ANY KEY TO CONTINUE

(n=4)

Read name

name=name of any contact or invalid

Yes → Paticular saved contact → ENTER ANY KEY TO CONTINUE

No → No contact added → ENTER ANY KEY TO CONTINUE

(n=5)

Read name

name=name of any contact or invalid

Yes → Delete that paticuar contact → CONTACT DELETED → ENTER ANY KEY TO CONTINUE

No → NO CONTACT → ENTER ANY KEY TO CONTINUE

(n=6)

EXIT THE PHONE BOOK

STOP

Pseudo code

START

-Initialize variables :

Char name,

char age ,

char gender,

long int mobile no,

char fathers name,

char email-id .

-Declare function

Void menu( )

void addrecord()

void listrecord()

void modifyrecord()

void deleterecord()

void searchrecord()

-Print "Welcome to phone book"

Print "Menu options"

1.Add New   2.List   3.Modify

4.Search   5.Delete   6.Exit

Read

int n

Switch 'n'

n=1:add record() break

n=2list record() break

n=3: modify record() break

n=4: search record() break

n=5: exit record() break

n=6: delete record() break

call menu()

FOR(n=1)

Read name, age, gender, father name, mobile no, email id

save the data in a file and

call menu()

For(n=2)

IF(file==NULL) THEN

PRINT error in opening

ELSE

Show the data saved in file

END IF

Call menu ()

FOR(n=3)

IF(file==NULL) THEN

PRINT error in opening

ELSE

Read name

IF(name=name of any contact) THEN

Read name, age, gender, father name, mobile no, email id

Update the contact information to this data

Save the updated data

ELSE

PRINT no contact added

END IF

Call menu()

FOR(n=4)

IF(file==NULL) THEN

PRINT error in opening

ELSE

Read name

IF(name=name of any contact) THEN

PRINT that particular contact details

ELSE

PRINT no contact added

END IF

FOR(n=5)

IF(file==NULL) THEN

PRINT error in opening

ELSE

Read name

IF(name=name of any contact) THEN

Delete the details of that contact from the file

ELSE

PRINT no contact added

END IF

Call menu()

FOR(n=6)

Exit the application

STOP

Code

```c
#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#include<windows.h>

struct person

{

char name[35];

char address[50];

char father_name[35];

long int mble_no;

char sex[8];

char mail[100];

};

void menu();

void got();

void start();

void back();

void addrecord();

void listrecord();

void modifyrecord();

void deleterecord();

void searchrecord();

int main()

{

system("color 5f");

start();
```

```c
return 0;

}

void back()

{

start();

}

void start()

{

menu();

}

void menu()

{

system("cls");

printf("\t\t*********WELCOME TO PHONEBOOK************");

printf("\n\n\t\t\t  MENU\t\t\n\n");

printf("\t1.Add New  \t2.List  \t3..Modify \n\t4.Search \t5.Delete \t6.Exit");

switch(getch())

{

case '1':

addrecord();

break;

case '2': listrecord();

break;

case '3': modifyrecord() ;

break;

case '4': searchrecord();

break;

case '5': deleterecord();

break;
```

```c
case '6': exit(0);

break;

default:

system("cls");

printf("\nEnter 1 to 6 only");

printf("\n Enter any key");

getch();

menu();

}

}

void addrecord()

{

system("cls");

FILE *f;

struct person p;

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

printf("\n Enter name: ");

got(p.name);

printf("\nEnter the address: ");

got(p.address);

printf("\nEnter father name: ");

got(p.father_name);

printf("\nEnter phone no.:");

scanf("%ld",&p.mble_no);

printf("Enter sex:");

got(p.sex);

printf("\nEnter e-mail:");

got(p.mail);

fwrite(&p,sizeof(p),1,f);
```

```c
fflush(stdin);

printf("\nrecord saved");

fclose(f);

printf("\n\nEnter any key");

getch();

system("cls");

menu();

}

void listrecord()

{

struct person p;

FILE *f;

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

if(f==NULL)

{

printf("\nfile opening error in listing :");

printf("\n\nEnter any key");

getch();

system("cls");

menu();

}

while(fread(&p,sizeof(p),1,f)==1)

{

printf("\n\n\n YOUR RECORD IS\n\n ");

printf("\nName=%s\nAdress=%s\nFathername=%s\nMobileno=%ld\nSex=%s\nEmail=%s",p.name,p.address,p.father_name,p.mble_no,p.sex,p.mail);

getch();

system("cls");

}
```

```c
fclose(f);

printf("\n Enter any key");

getch();

system("cls");

menu();

}

void searchrecord()

{

struct person p;

FILE *f;

char name[100];

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

if(f==NULL)

{

printf("\n error in opening\a\a\a\a");

printf("\n\nEnter any key");

getch();

system("cls");

menu();

}

printf("\nEnter name of person to search\n");

got(name);

while(fread(&p,sizeof(p),1,f)==1)

{

if(strcmp(p.name,name)==0)

{

printf("\n\tDetail Information About %s",name);

printf("\nName:%s\naddress:%s\nFathername:%s\nMobileno:%ld\nsex:%s\nEmail:%s",p.name,p.address,p.father_name,p.mble_no,p.sex,p.mail);
```

```c
}
else
printf("file not found");
}
fclose(f);
printf("\n Enter any key");
getch();
system("cls");
menu();
}
void deleterecord()
{
struct person p;
FILE *f,*ft;
int flag;
char name[100];
f=fopen("project","rb");
if(f==NULL)
{
printf("\nCONTACT'S DATA NOT ADDED YET.");
}
else
{
ft=fopen("temp","wb+");
if(ft==NULL)
printf("file opaning error");
else
{
printf("\nEnter CONTACT'S NAME:");
```

```c
got(name);

fflush(stdin);

while(fread(&p,sizeof(p),1,f)==1)

{

if(strcmp(p.name,name)!=0)

fwrite(&p,sizeof(p),1,ft);

if(strcmp(p.name,name)==0)

flag=1;

}

fclose(f);

fclose(ft);

if(flag!=1)

{

printf("\nNO CONACT'S RECORD TO DELETE.");

remove("temp.txt");

}

else

{

remove("project");

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

printf("\nRECORD DELETED SUCCESSFULLY.");

}

}

}

printf("\n Enter any key");

getch();

system("cls");

menu();

}
```
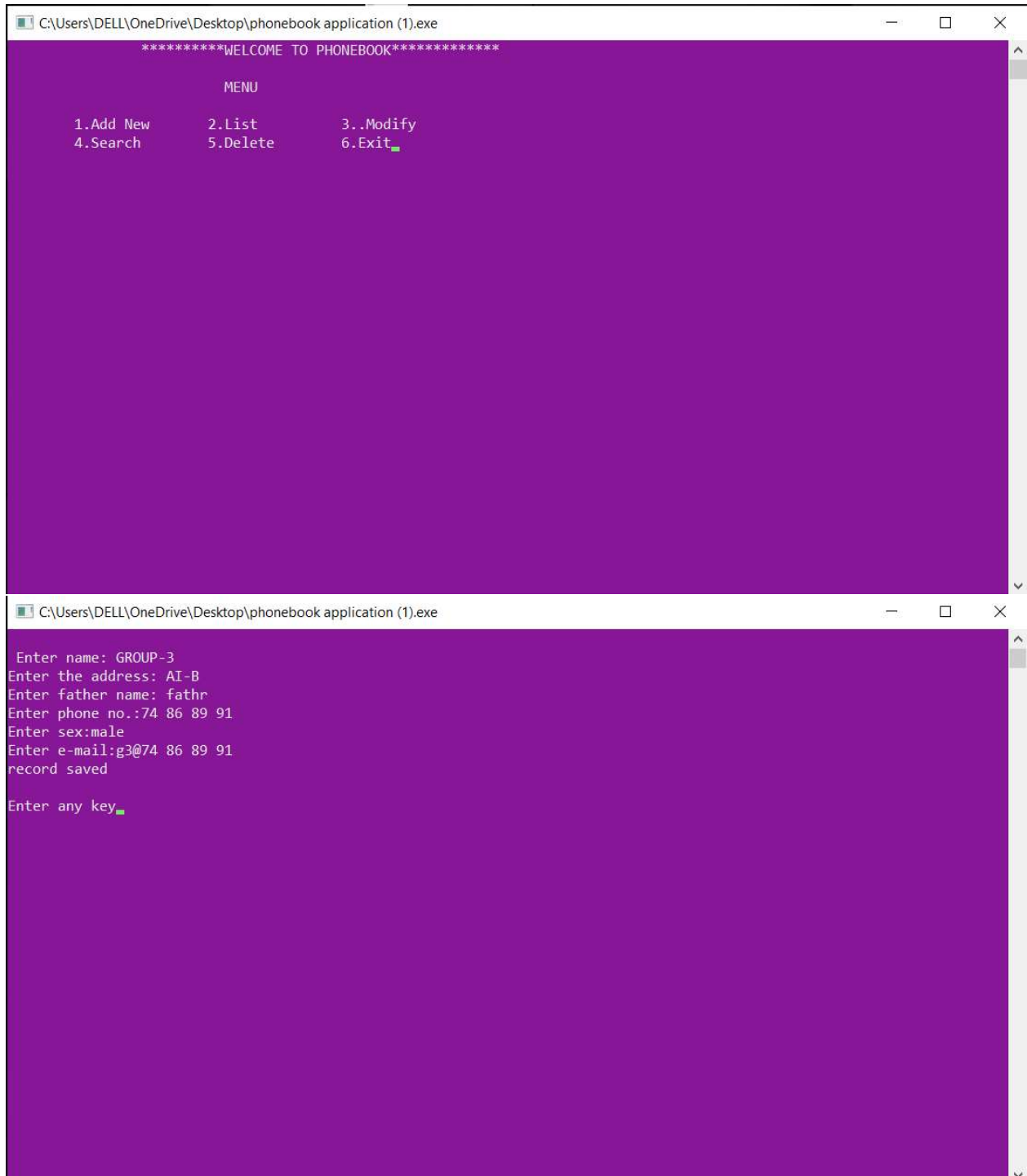
```c
void modifyrecord()
{
int c;
FILE *f;
int flag=0;
struct person p,s;
char  name[50];
f=fopen("project","rb+");
if(f==NULL)
{
printf("\nCONTACT'S DATA NOT ADDED YET.");
printf("\n\nEnter any key");
getch();
system("cls");
menu();
}
else
{
system("cls");
printf("\nEnter CONTACT'S NAME TO MODIFY:\n");
got(name);
while(fread(&p,sizeof(p),1,f)==1)
{
if(strcmp(name,p.name)==0)
{
printf("\n Enter name:");
got(s.name);
printf("\nEnter the address:");
got(s.address);
```

```c
printf("\nEnter father name:");

got(s.father_name);

printf("\nEnter phone no:");

scanf("%ld",&s.mble_no);

printf("\nEnter sex:");

got(s.sex);

printf("\nEnter e-mail:");

got(s.mail);

fseek(f,-sizeof(p),SEEK_CUR);

fwrite(&s,sizeof(p),1,f);

flag=1;

break;

}

fflush(stdin);

}

if(flag==1)

{

printf("\n your data id modified");

}

else

{

printf(" \n data is not found");

}

fclose(f);

}

printf("\n Enter any key");

getch();

system("cls");

menu();
```

```c
}
void got(char *name)
{
int i=0,j;
char c,ch;
do
{
c=getch();
if(c!=8&&c!=13)
{
*(name+i)=c;
putch(c);
i++;
}
if(c==8)
{
if(i>0)
{
i--;
}
// printf("h");
system("cls");
for(j=0;j<i;j++)
{
ch=*(name+j);
putch(ch);
}
}
}while(c!=13);
```

```
*(name+i)='\0';

}
```

```
          **********WELCOME TO PHONEBOOK*************

                    MENU

         1.Add New      2.List        3..Modify
         4.Search       5.Delete      6.Exit


 YOUR RECORD IS


Name=GROUP-3
Adress=AI-B
Father name=fathr
Mobile no=74
Sex=male
E-mail=g3@74 86 89 91
```

```
Enter CONTACT'S NAME TO MODIFY:
GROUP-3
 Enter name:zethu pavan sidhart tharun
Enter the address:srm ai-b
Enter father name:father
Enter phone no:74 91 89 86

Enter sex:male
Enter e-mail:g3@457812
 your data id modified
 Enter any key
```

```
 C:\Users\DELL\OneDrive\Desktop\phonebook application (1).exe                          —    □    ✕

          **********WELCOME TO PHONEBOOK*************

                    MENU

        1.Add New       2.List        3..Modify
        4.Search        5.Delete      6.Exit
Enter name of person to search
zethu pavan sidhart tharun
        Detail Information About zethu pavan sidhart tharun
Name:zethu pavan sidhart tharun
address:srm ai-b
Father name:father
Mobile no:74
sex:male
E-mail:g3@457812
 Enter any key_
```

Conclusion :

From this phone book application we can learn about how manage files and structures in c language and this is very helpful to understand about c language and it's properties for beginners. Not only files we can also cover different parts of c language like looping arrays and all. This project helped us to understand about c language.

THE END