

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590 018



2nd YEAR C_PRACTICING PROJECT WORK REPORT
on

“Phonebook Program”

SUBMITTED BY

Tejaswini B M 1RN19CS167

Under the Guidance of

Mrs. Seema H R
Assistant Professor



Department of Computer Science and Engineering
(NBA Accredited for academic years 2018-19, 2019-20, 2020-21, 2021-22)

R N S Institute of Technology
Channasandra, Dr.Vishnuvardan Road, Bengaluru-560 098
2020-2021

R N S Institute of Technology
Channasandra, Dr.Vishnuvardan Road, Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(NBA ACCREDITED FOR ACADEMIC YEARS 2018-19, 2019-20, 2020-21, 2021-22)



CERTIFICATE

Certified that the **C Practicing Project** work entitled “ **Phonebook Program** ” has been successfully carried out by **Tejaswini B M** bearing USN 1RN19CS167, a bonafide student of **RNS Institute of Technology**, currently studying 2nd year **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2020-21. It is certified that all corrections/suggestions indicated by **Mrs. Seema H R** have been incorporated in the report deposited in the departmental library.

Mrs. Seema H R
Assistant Professor

Dr. Kiran P
Professor & HOD

Dr. M K Venkatesha
Principal

ABSTRACT

Phonebook in C is a console application without graphics. This program deals with combination of structures, arrays and file handling functions. Phonebook is a collection of contact details. The basic contact details include name, address, phone number and email id.

We can perform basic operations like insertion of contacts, deletion of contacts, modification of already entered contact details and search for contact name and retrieve contact details. File handling functions in C enables us to create, update, read, and delete the files stored on the local file system through our C program. Hence file handling functions along with basic data structures is one of the easy approach to implement phonebook program.

ACKNOWLEDGMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work.

I would like to take this opportunity to thank them all. I would like to thank Dr. M K Venkatesha, Principal, RNSIT, Bangalore, for his support towards completing this mini project. I would like to thank Dr. Kiran P, Prof. & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice. I deeply express sincere gratitude to my mentor Mrs. Seema H R, Department of CSE, RNSIT, Bangalore, for her/his able guidance, regular source of encouragement and assistance throughout this project. I also would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date : 21/07/2021

Place : Bengaluru

Tejaswini B M

1RN19CS167

TABLE OF CONTENTS

Sl No.	Chapters Name	Page No.
1.	INTRODUCTION	1
	1.1 PROBLEM STATEMENT	1
	1.2 ADVANTAGES OF PROPOSED SYSTEM	1
2.	REQUIREMENT	2
	2.1 SOFTWARE REQUIREMENT	2
3.	SYSTEM DESIGN	3
	3.1 FLOW CHART	3
4.	IMPLEMENTATION	7
	4.1 CODE	7
5.	RESULT ANALYSIS	16
	5.1 SAMPLE INPUTS OUTPUTS	16
6.	CONCLUSION AND FUTURE WORK	21
	6.1 CONCLUSION	21
	6.2 LIMITATIONS AND FUTURE WORK	21
7.	BIBLIOGRAPHY	22

CHAPTER 1

INTRODUCTION

Phonebook in C is a console application without graphics. Phonebook is a collection of contact details. The basic contact details include name, address, phone number and email id. We can perform basic operations like insertion of contacts, deletion of contacts, modification of already entered contact details and search for contact name and retrieve contact details. File handling functions in C enables us to create, update, read, and delete the files stored on the local file system through our C program.

The following operations can be performed on a file.

- 1) Creation of the new file
- 2) Opening an existing file
- 3) Reading from the file
- 4) Writing to the file
- 5) Deleting the file
- 6) Searching the file

1.1 PROBLEM STATEMENT

To develop a Phonebook Management program in C making use of Data Structures and File Handling Functions.

1.2 ADVANTAGES OF PROPOSED SYSTEM

The advantages of Phonebook program are as follows:

- 1) It becomes easier to store extra details like email, address along with contact name itself.
- 2) It is easy to retrieve entire contact details by just entering name.
- 3) Deletion and modification of contact detail is simple.
- 4) Code is portable i.e, can run on any system without any difficulties.

CHAPTER 2

REQUIREMENT

2.1 SOFTWARE REQUIREMENT

TOOLS :

TURBO C a light weighted tool which can be installed in any kind of window OS. Turbo C is much simpler to configure & use, runs on old DOS machines. Also it is compact in size. Hence we used this for implementation.

ONLINE GDB is a compiler and debugger tool for C/C++ languages. It is world's first online IDE which gives debugging facility with embedded GDB debugger. This is very handy web app for coders who loves coding in online IDE. This is more compactible platform for compiling and execution of program.

PROGRAMMING LANGUAGE :

C being a general purpose and structured oriented language, has wide variety of applications, which includes scripting the operating system and embedded systems like micro controllers since C code are easily portable. UNIX was the first OS to be scripted in C, then later MS Windows was scripted. C is also used in design of compiler and for gaming purpose also. It is also used in photoshop and other editors.

C was initially used for system development work, in particular, the programs that make-up the operating system. C is used mainly because it produces code that runs nearly as fast as code written in assembly language. Some examples of the use of C might be: Operating Systems, Language Compilers, Assemblers, Text Editors, Print Spoolers, Network Drivers, Modern Programs, Data Bases, Language Interpreters, Utilities.

OPERATING SYSTEM :

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. One such operating system is Windows 10.

WINDOWS 10 provides desktop oriented interface. It is a major release of the Windows NT operating system developed by Microsoft. It is the successor to Windows 8.1, released nearly two years earlier, and was released to manufacturing on July 15, 2015, and broadly released for the general public on July 29, 2015.

CHAPTER 3

SYSTEM DESIGN

A flowchart is a diagrammatic representation of an algorithm. It is the most widely used graphical representation of an algorithm and procedural design workflows. It flows in sequential order and briefs us how control flows. The diagrammatic representation for phonebook program is illustrated below.

3.1 FLOWCHART OF PHONEBOOK PROGRAM

3.1.1 FLOWCHART OF MAIN FUNCTION

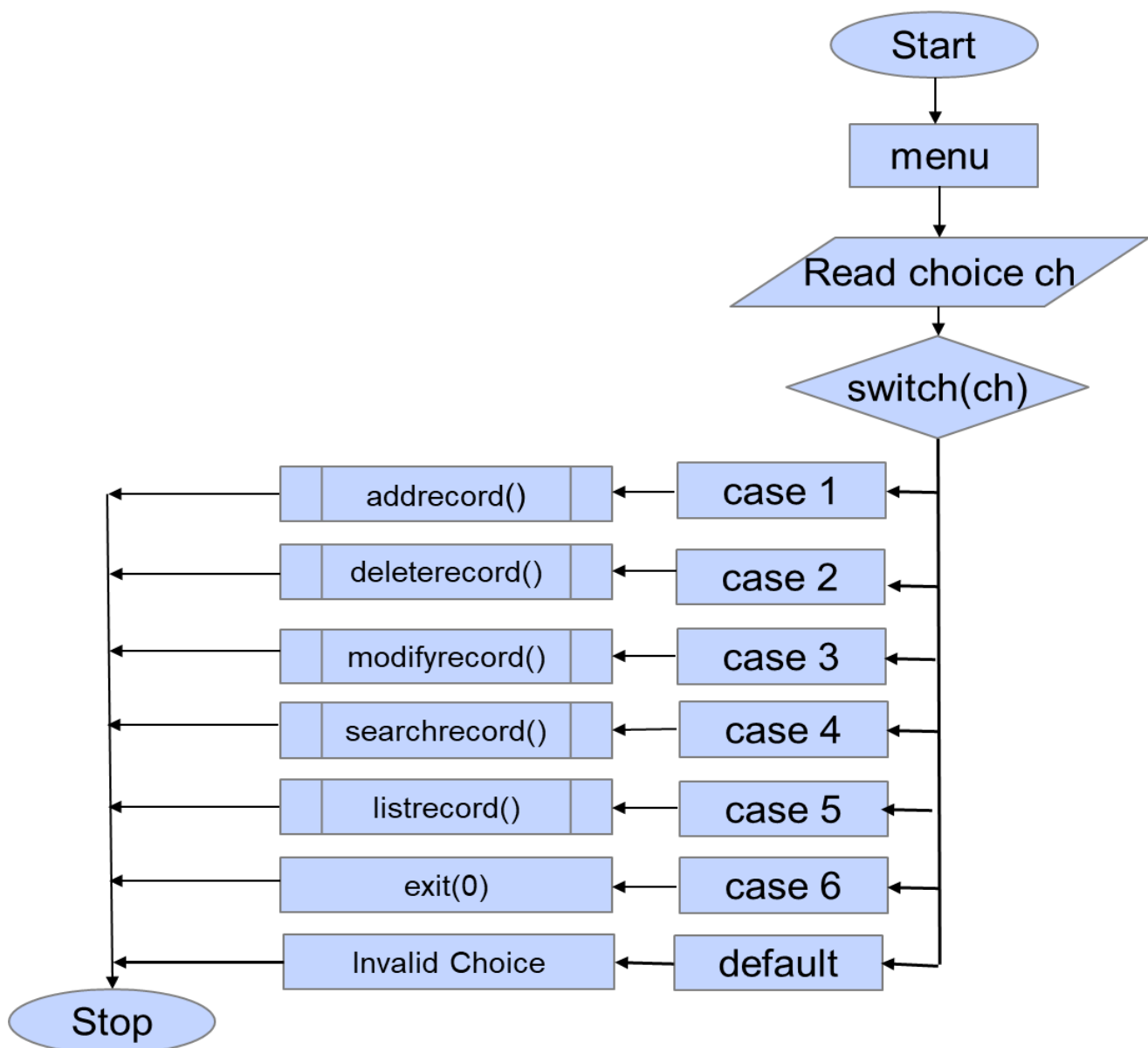


Fig 3.1.1 Flowchart of main()

The main() function is where actual program execution begins. Inside this function menu() is called. In menu(), we have switch statement in which each cases corresponds to operation performed. Particular case is selected based on entered choice. Then the control goes to that function for performing the operation.

3.1.2 FLOWCHART OF ADD FUNCTION

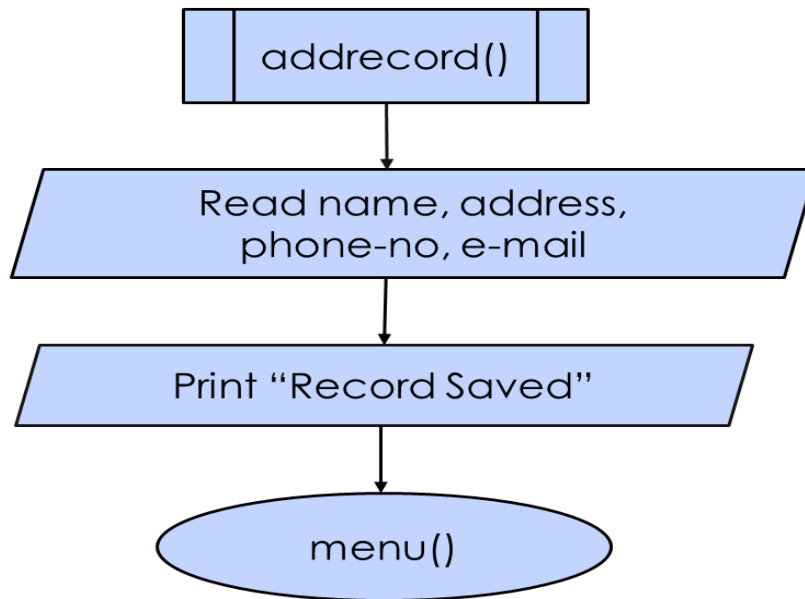


Fig 3.1.2 Flowchart of `addrecord()`

If choice 1 is selected, then `addrecord()` function gets executed. Here user is asked to enter details like name, address, phone-no and e-mail. Then message "Record saved" is printed.

3.1.3 FLOWCHART OF DELETE FUNCTION

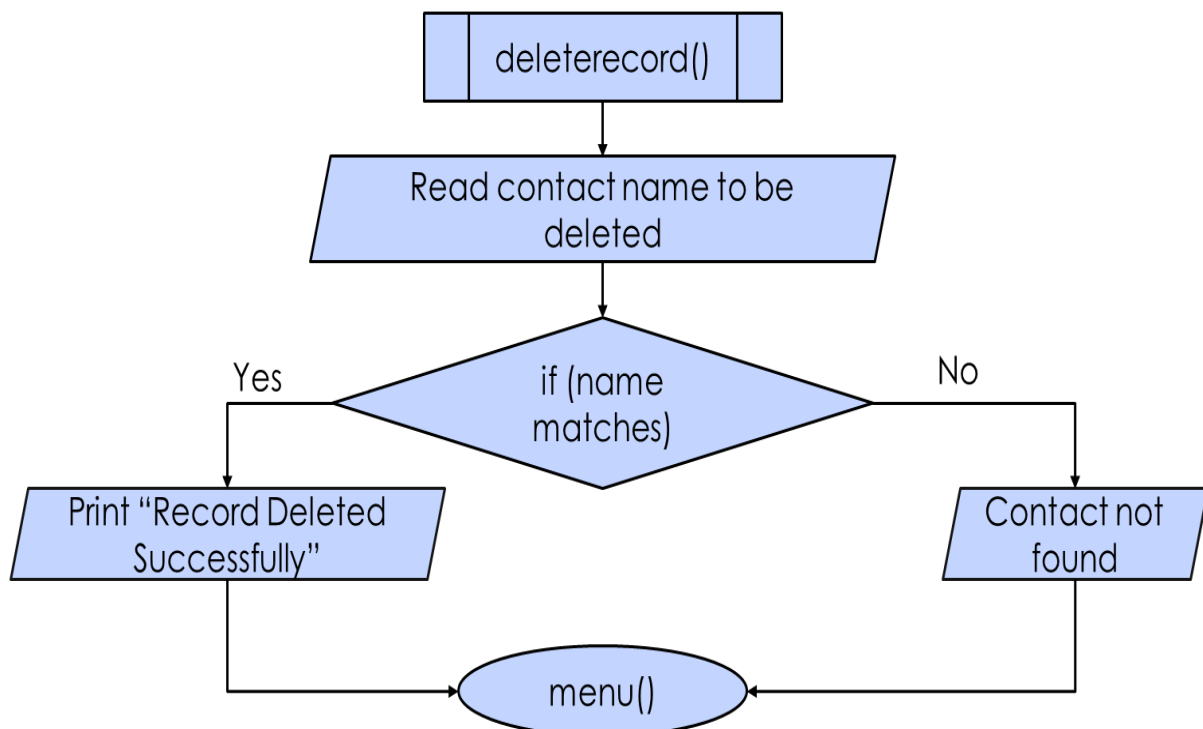


Fig 3.1.3 Flowchart of `deleterecord()`

If choice 2 is selected, then `deleterecord()` function gets executed. Here user is asked to enter name. If the name matches with the name in record, then message "Record deleted successfully" is printed or else "Contact not found" is printed.

3.1.4 FLOWCHART OF MODIFY FUNCTION

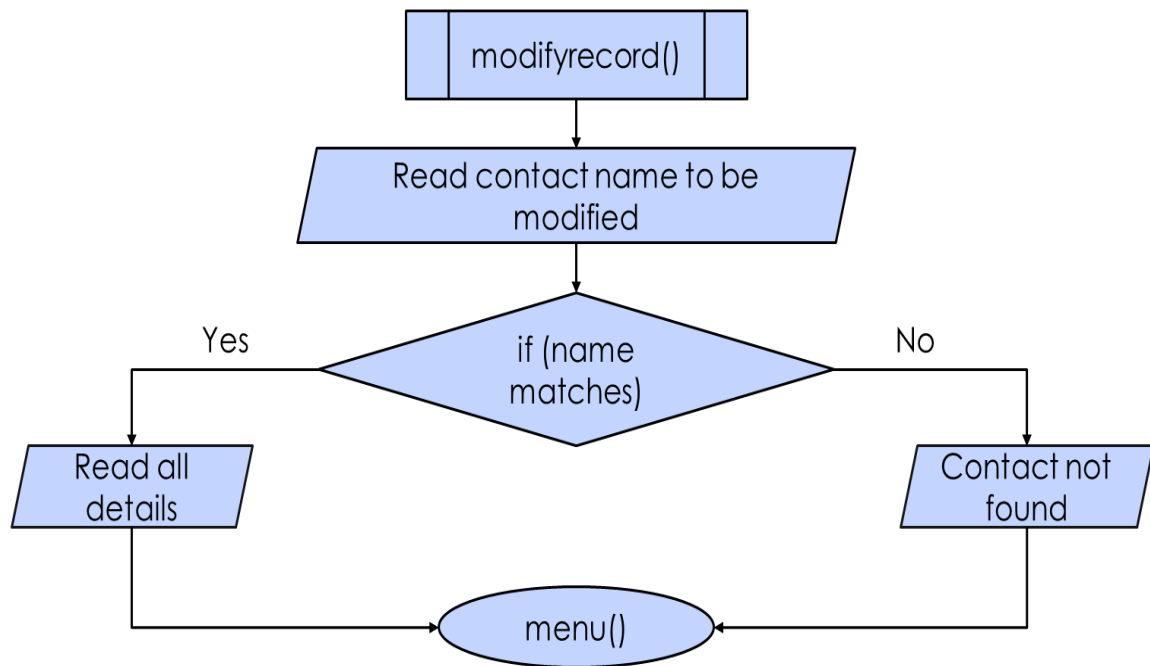


Fig 3.1.4 Flowchart of `modifyrecord()`

If choice 3 is selected, then `modifyrecord()` function gets executed. User has to enter name. If the name matches with the name in record, then all details need to be entered by user. Else "Contact not found" is printed.

3.1.5 FLOWCHART OF SEARCH FUNCTION

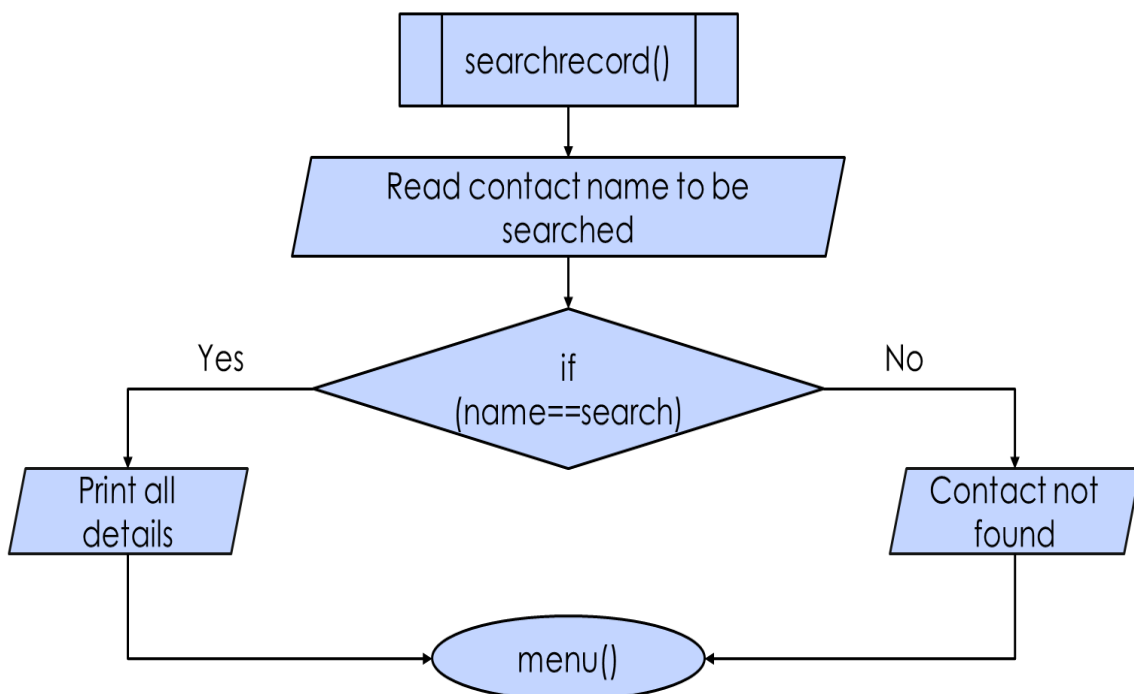


Fig3.1.5 Flowchart of `searchrecord()`

If choice 4 is selected, then `searchrecord()` function gets executed. Here user is asked to enter name. If the name matches with the name in record, then all details will be printed. Else "Contact not found" is printed.

3.1.6 FLOWCHART OF LIST FUNCTION

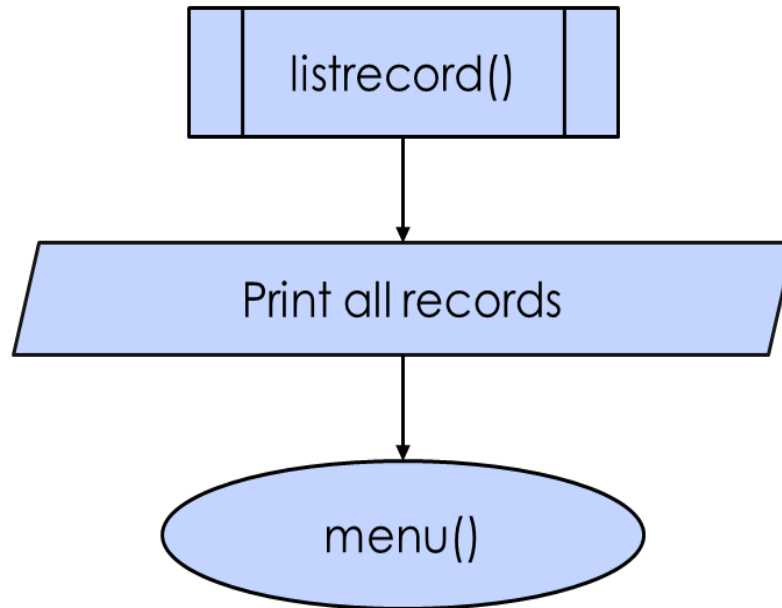


Fig 3.1.6 Flowchart of listrecord()

If choice 5 is selected, then `listrecord()` function gets executed. All the details(name, address, phone-no and e-mail) of all records in the file is printed in the order as entered by user.

CHAPTER 4

IMPLEMENTATION

4.1 CODE

```
//Phonebook program using C
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

//Function Declarations
void addrecord();
void deleterecord();
void modifyrecord();
void searchrecord();
void listrecord();
int menu();

//creating user-defined data type
struct person
{
    char name[35];
    char address[50];
    char mble_no[20];
    char mail[100];
};

typedef struct person ch; //ch is a name(alias name) to already existing type
ch p; //creating variable p of type struct person

//function to insert contact details
void addrecord()
{
    FILE *f, *ft;
    char name[100];
    f=fopen("project","ab+"); // opens a new file project in read and write in binary format
    printf("\nEnter name: ");
```

```
scanf("%s",name);
fflush(stdin); //to clear (or flush) the output buffer and move the buffered data to console
if(f!=NULL)
{
    ft=fopen("temp","wb+"); //opens a binary file in read and write mode
    {
        if(strcmp(p.name,name)==0) //if strings are equal that is name already exists in file
        {
            printf("Name already exists\nKindly enter another name:");
            scanf("%s",p.name);
        }
        else
        {
            strcpy(p.name,name); //copy name to p variable
        }
    }
}
printf("\nEnter the address: ");
scanf("%s",p.address);
char mobno[20]="";
while(1)
{
    printf("\nEnter 10 digit phone no:");
    scanf ("%s",mobno);
    if(strcmp(p.mble_no,mobno)==0) //ph num already exists in file
    {
        printf("Contact already exists\nKindly check.");
        //scanf("%s",mobno);
        continue;
    }
    if(strlen(mobno)<= 0)//ph num not entered
    {
        printf("Please enter ph num\n");
        continue;
    }
}
```

```
    if(atoi(mobno)<= 0)// if character entered in ph num
    {
        printf("Character not allowed.\n");
        continue;
    }

    if(strlen(mobno)< 10 || strlen(mobno)> 10) //ph num less than 10 digits
    {
        printf("Only 10 digit is allowed\n");
        continue;
    }
    strcpy(p.mble_no, mobno);
    break;
}
while(1)
{
    char mail[50];
    //only e-mail ID of these formats is allowed (@gmail.com/@yahoo.com/@hotmail.com)
    printf ("\nEnter e-mail ID (@gmail.com/@yahoo.com/@hotmail.com):");
    scanf ("%s",mail);
    char a[25]="@gmail.com";
    char b[25]="@yahoo.com";
    char c[25]="@hotmail.com";
    char gmail[50]="";
    char yahoo[50]="";
    char hotmail[50]="";
    char *z, *q;
    if( (z=strstr(mail, "@gmail.com")) != NULL || (z=strstr(mail, "@yahoo.com")) != NULL ||
(z=strstr(mail, "@hotmail.com")) != NULL)
    {
        strcpy (p.mail, mail);
        goto label;
    }
    else
    {
        printf("Invalid mail format\n");
```

```
printf("kindly use @gmail.com/@yahoo.com/@yahoo.com\n");

        continue;
    }
}

label:
fwrite(&p,sizeof(p),1,f); // writes 1(count) ele, each of size p from block of mry pointed by p
fflush(stdin);
printf("\nRecord saved");
fclose(ft);
fclose(f); //closes the f file created
menu();
}

//function to delete contact details
void deleterecord()
{
    FILE *f,*ft;
    int flag;
    char name[100];
    f=fopen("project","rb+");//opens a new file project in read and write in binary format
    if(f!=NULL)
    {
        ft=fopen("temp","wb+");//create temp file and allow write and read operation on file
        {
            printf("Enter Contact Name:");
            scanf("%s",name);
            fflush(stdin);
            while(fread(&p,sizeof(p),1,f)==1)// fread() function is used to read bytes form the file
            {
                if(strcmp(p.name,name)!=0)
                    fwrite(&p,sizeof(p),1,ft); //fwrite() function is used to write records (sequence of bytes) to the file
                if(strcmp(p.name,name)==0)
                    flag=1; //flag is set to 1 if contact name is found to perform deletion
            }
            fclose(f);
```

```
fclose(ft);
if(flag!=1)
{
printf("Invalid contact name");
remove("temp.txt");
}
else
{
remove("project");
rename("temp","project");
printf("Record Deleted Successfully\n");
}
}
}
else
{
printf("Contact is empty\n");
}
menu();
}
```

//function to list all contacts in file in the order entered by user

```
void listrecord()
{
FILE *f;
char *temp;
f=fopen("project","rb");// opens a new file project in read mode in binary format
if(f!=NULL)
{
printf("-----\n");
printf("\t\tPhonebook Record\n");
printf("-----");
printf("\nName\t\tAddress\t\tPhone Number\t\tMail_Id\n");
printf("-----\n");
while(fread(&p,sizeof(p),1,f)==1)
{

```



```
    printf("%s\t\t%s\t\t%s\t\t%s\n",p.name,p.address,p.mble_no,p.mail);
}
printf("-----\n");
}
else
{
    printf("No Records\n");
}
menu();
}
```

//function to modify contact details

void modifyrecord()

```
{
    FILE *f;
    int flag=0;
    struct person p,s;
    char name[50];
    f=fopen("project","rb+");// opens a new file project in read and write in binary format
    if(f==NULL)
    {
        printf("Contact is empty\n");
    }
    else
    {
        printf("\nEnter contact name to modify:\n");
        scanf("%s",name);
        while(fread(&p,sizeof(p),1,f)==1)
        {
            if(strcmp(name,p.name)==0)// if entered name matches with names in file then modify contact
            {
                printf("\nEnter name:");
                scanf("%s",s.name);
                printf("\nEnter the address: ");
                scanf("%s",s.address);
                char mobno[20]="";
```

```
while(1)
{
    printf("\nEnter 10 digit phone no:");
    scanf ("%s",mobno);
    if(strlen(mobno)<= 0)
    {
        printf("Please enter ph num\n");
        continue;
    }
    if(atoi(mobno)<= 0)
    {
        printf("Character not allowed.\n");
        continue;
    }

    if(strlen(mobno)< 10 || strlen(mobno)> 10)
    {
        printf("Only 10 digit is allowed\n");
        continue;
    }
    strcpy(s.mble_no, mobno);
    break;
}
printf ("\nEnter e-mail ID (@gmail.com/@yahoo.com/@hotmail.com):");
    scanf ("%s",s.mail);
    fseek(f,-sizeof(p),SEEK_CUR);
    fwrite(&s,sizeof(p),1,f);
    flag=1;//flag is set to 1 if contact is modified
    break;
}
fflush(stdin);

}
if(flag==1)
{
    printf("\nContact is modified");
```

```
    }
    else
    {
        printf("\nContact not found");
    }
    fclose(f);
}
menu();
}

//function to search contact name and retrieve contact details
void searchrecord()
{
    char name[100];
    FILE *f;
    f=fopen("project","rb+");//opens a new file project in read and write in binary format
    if(f==NULL)//if file is null
    {
        printf("\nContact is empty");
        exit(1);
    }
    else
    {
        printf("\nEnter contact name to search:");
        scanf("%s",name);
        if(strcmp(p.name,name)!=0)//if name doesn't match with names in file
        {
            printf("Contact not found\n");
        }
        else
        {
            printf("\nContact Details About %s",name);//print all details of that contact
            printf("\nName:%s\nAddress:%s\nMobile no:%s\nE-mail:%s\n",p.name,p.address,p.mble_no,p.mail);
        }
    }
    menu();
}
```

//menu() contains menu of program with each case assigned with operations in switch statement.

```
int menu()
{
    int choice;
    printf("\n-----PHONEBOOK PROGRAM-----\n");
    printf("\n\n\t\t\t\t\t MENU\t\t\n\n");
    printf("1.Add Contact\t2.Delete Contact\t3.Modify Contact\t4.Search Contact\t5.List Contacts\t6.Exit");
    printf("\n-----\n");
    printf("\nEnter the choice:");
    scanf("%d",&choice);
    for(;;)
    {
        switch(choice)
        {
            case 1 : addrecord();break;
            case 2 : deleterecord();break;
            case 3 : modifyrecord();break;
            case 4 : searchrecord();break;
            case 5 : listrecord();break;
            case 6 : exit(0);break;
            default: printf("Invalid choice\nPlease enter valid choice:");
        }
        return (0);
    }
}

//execution of program begins here
int main()
{
    menu();//function call
    return(1);//function is returning true.
}
```

CHAPTER 5

RESULT ANALYSIS

5.1 SAMPLE INPUTS OUTPUTS

When no contact is saved in file, but the user wants to perform operations. Then below output is expected.

```
-----PHONEBOOK PROGRAM-----  
                                MENU  
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit  
-----  
Enter the choice:2  
Contact is empty  
  
-----PHONEBOOK PROGRAM-----  
                                MENU  
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit  
-----  
Enter the choice:3  
Contact is empty  
  
-----PHONEBOOK PROGRAM-----  
                                MENU  
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit  
-----  
Enter the choice:4  
Contact is empty  
  
-----PHONEBOOK PROGRAM-----  
                                MENU  
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit  
-----  
Enter the choice:5  
No Records
```

Fig 5.1.1 Contact is Empty

When user enters invalid choice, then below output is expected.

```
-----PHONEBOOK PROGRAM-----  
                                MENU  
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit  
-----  
Enter the choice:8  
Invalid choice  
Please enter valid choice:
```

Fig 5.1.2 Illustration of invalid choice

Insertion of 5 contacts into file record with demonstration of invalid type like (mail format, ph-no).

```
-----PHONEBOOK PROGRAM-----  
  
                        MENU  
  
1.Add Contact    2.Delete Contact    3.Modify Contact    4.Search Contact    5.List Contacts 6.Exit  
-----  
Enter the choice:1  
Enter name: Anusha  
Enter the address: Bidadi  
Enter 10 digit phone no:98473  
Only 10 digit is allowed  
Enter 10 digit phone no:hsaju2232  
Character not allowed.  
Enter 10 digit phone no:9164071399  
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):anu123@email.com  
Invalid mail format  
kindly use @gmail.com/@yahoo.com/@yahoo.com  
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):anu123@gmail.com  
Record saved  
-----PHONEBOOK PROGRAM-----  
  
                        MENU  
  
1.Add Contact    2.Delete Contact    3.Modify Contact    4.Search Contact    5.List Contacts 6.Exit  
-----  
Enter the choice:1  
Enter name: Banu  
Enter the address: Mysore  
Enter 10 digit phone no:9164071399  
Contact already exists  
Kindly check.  
Enter 10 digit phone no:9980988321  
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):banu@hotmail.com  
Record saved  
-----PHONEBOOK PROGRAM-----  
  
                        MENU  
  
1.Add Contact    2.Delete Contact    3.Modify Contact    4.Search Contact    5.List Contacts 6.Exit  
-----  
Enter the choice:1  
Enter name: Frank  
Enter the address: Kengeri  
Enter 10 digit phone no:9876543210  
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):frank@yahoo.com  
Record saved  
-----PHONEBOOK PROGRAM-----  
  
                        MENU  
  
1.Add Contact    2.Delete Contact    3.Modify Contact    4.Search Contact    5.List Contacts 6.Exit  
-----  
Enter the choice:1  
Enter name: Govind  
Enter the address: Bidadi  
Enter 10 digit phone no:9980976521  
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):gov12@gmail.com  
Record saved
```

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:1
Enter name: Sirish
Enter the address: Mysore
Enter 10 digit phone no:99876504321
Only 10 digit is allowed
Enter 10 digit phone no:9976543201
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):sirish@gmail.com
Record saved
  
```

Fig 5.1.3 Insertion of Contact

After insertion, contacts in file are shown below.

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:5
-----
Phonebook Record
-----
Name           Address        Phone Number    Mail_Id
-----
Anusha         Bidadadi       9164071399      anu123@gmail.com
Banu           Mysore         9980988321      banu@hotmail.com
Frank          Kengeri        9876543210      frank@yahoo.com
Govind         Bidadadi       9980976521      gov12@gmail.com
Sirish         Mysore         9976543201      sirish@gmail.com
-----
  
```

Fig 5.1.4 List of Contact Details in File

Case1: In searching of contact, contact name not found in file

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:4
Enter contact name to search:Nayana
Contact not found
  
```

Case2: In searching of contact, contact name found and details is printed

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:4
Enter contact name to search:Sirish
Contact Details About Sirish
Name:Sirish
Address:Mysore
Mobile no:9976543201
E-mail:sirish@gmail.com
  
```

Fig 5.1.5 Searching for a Contact

Modification of contact

Case1: Contact name not found in file

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:3
Enter contact name to modify:
anu
Contact not found
    
```

Case2: Contact name found and modified details are entered

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:3
Enter contact name to modify:
Frank
Enter name:Emily
Enter the address: Mysore
Enter 10 digit phone no:9854763210
Enter e-mail ID(@gmail.com/@yahoo.com/@hotmail.com):emily@gmail.com
Contact is modified
    
```

Fig 5.1.6 Modifying Contact Details

After modification contacts in the file is shown below.

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:5
-----
Phonebook Record
-----
Name           Address        Phone Number    Mail_Id
-----
Anusha         Bidadi         9164071399      anu123@gmail.com
Banu           Mysore         9980988321      banu@hotmail.com
Emily          Mysore         9854763210      emily@gmail.com
Govind         Bidadi         9980976521      gov12@gmail.com
Sirish         Mysore         9976543201      sirish@gmail.com
-----
    
```

Fig 5.1.7 List of Contact Details in File

Deletion of contact

Case1: Contact name not found in file

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:2
Enter Contact Name:Rahul
Invalid contact name
    
```

Case2: Contact name found in file then delete entire details of that contact

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:2
Enter Contact Name:Govind
Record Deleted Successfully
    
```

Fig 5.1.8 Deletion of Contact

After deletion contacts in the file record is shown below,

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:5
-----
Phonebook Record
-----
Name           Address      Phone Number   Mail_Id
-----
Anusha         Bidadi       9164071399    anu123@gmail.com
Banu           Mysore       9980988321    banu@hotmail.com
Emily          Mysore       9854763210    emily@gmail.com
Sirish         Mysore       9976543201    sirish@gmail.com
-----
    
```

Fig 5.1.9 List of Contact Details in File

To exit from phonebook program, choice 6 need to be entered

```

-----PHONEBOOK PROGRAM-----
                                MENU
1.Add Contact   2.Delete Contact   3.Modify Contact   4.Search Contact   5.List Contacts 6.Exit
-----
Enter the choice:6
    
```

Fig 5.1.10 Closing of phonebook program

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Phonebook program is user friendly and easy to manage and handle contact details. This program allows user to insert, delete, modify and search contacts based on user choice. Each operation is made as an individual function and hence control goes to each block for execution.

6.2 LIMITATION AND FUTURE WORK

This phonebook program doesn't work properly, if contacts exceed over 200. And it cannot list the contacts in alphabetical order. In future we like to work on this.

CHAPTER 7

BIBLIOGRAPHY

- [1] Ellis Horowitz and Sartaj Sahni “Fundamentals of Data Structures in C” , 2nd Edition , University Press,2014
- [2] Seymour Lipschutz, “Data Structures Schaum’s Outlines”, Revised 1st Edition, McGraw Hill,2014
- [3] <https://intellipaat.com/blog/tutorial/c-tutorial/file-handling-c/>
- [4] <https://www.geeksforgeeks.org/basics-file-handling-c/>
- [5] <https://www.javatpoint.com/file-handling-in-c>
- [6] <https://www.tutorialspoint.com/basics-of-file-handling-in-c-programming>
- [7] <https://data-flair.training/blogs/file-handling-in-c/>
- [8] <https://www.w3schools.in/c-tutorial/file-handling/>
- [9] <https://www.guru99.com/c-file-input-output.html>
- [10] <https://fresh2refresh.com/c-programming/c-file-handling/>