**PHONEBOOK APPLICATION**

**ABSTRACT:**

Phone book application is primarily meant for keeping the records of the persons. Phone book application will provide the basic set of features of adding a new contact, searching, updating, deleting a contact .This mini project in C Phonebook allows to perform simple Phonebook operations like in the mobile. One 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. The source code is complete and totally error-free. It is compiled in Code Blocks with GCC compiler. Functions, file handling and data structure are used. This application contains how to add, list, modify or edit, search and delete data to/from the file. 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 .Personal information such as name, gender, 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.

**INDEX**

**S.NO TITLE P.NO**

**1. INTRODUCTION**

**2. PROGRAMMING LANGUAGE**

**3. FLOW CHART**

**4. SOURCE CODE**

**5. OUTPUT**

**6. CONCLUSION**

**7. REFERENCES**

**CHAPTER I**

* 1. **INTRODUCTION**

Phonebook is a very simple mini project in C that can help you understand the basic concepts of functions, file handling and data structure. This application will teach you how to add, list, modify or edit, search and delete data to/from the file.

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. It is estimated that there are more than 600 million mobile phone users around the world – and the number is increasing. The success of mobile phones seems easy to explain: being available anytime and anywhere. If a businessman moves from one place to another he is available for his company at no extra effort. If the son will be late at home he can inform his parents. If you have trouble with your car you can call the breakdown service. Mobile phones are used for communication but also for coordination between people. Informal meetings between people (e.g. going to a pub) are often not planned anymore in advance – they are arranged on time by mobile communication. Altogether there are a lot of opportunities using mobile phones. But there are also some emerging problems. Being reachable always and everywhere is solved by technology – but being reachable does not necessarily mean being available to all potential callers! Sharing information about the real world context of users is a solution to this problem . But until now the user interfaces of mobile phones are still very much based on legacy. In particular UIs and standard applications are not context dependent and therefore the caller has very little chance to know when making a call in what situation the called party will be interrupted.

The main idea of the program is to increase the awareness of working with computer systems and utilize computer resources for better result orientation. The main target behind the creation of this package is to remove the pressure from the user to learn about computer & software. It helps to maintain an automatic communication between the computer and the user. This Program “My Phonebook” facility the user with different records can add and search easily by using Button tools & Search options. There are some features also like: - Viewing Data in Data grid, and Logout. Our Project has following :- Login form “ADD” button for adding data in Database. “DELETE” button for deleting data in Database. “UPDATE” button for updating data in Database. “SEARCH” button for searching data in Database.

**PROGRAMMING LANGUAGE-C**

C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs. C was originally first implemented on the DEC PDP-11 computer in 1972.

In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard.

The UNIX operating system, the C compiler, and essentially all UNIX application programs have been written in C. C has now become a widely used professional language for various reasons −

* Easy to learn.
* Structured language.
* It produces efficient programs.
* It can handle low-level activities
* It can be compiled on a variety of computer platforms.



**DENNIS RITCHE –FOUNDER OF C LANGUAGE**

C has been used successfully for every type of programming problem imaginable from operating systems to spread sheets to expert systems - and efficient [**compilers**](http://www.le.ac.uk/cc/glossary/ccglc.html#47) are available for machines ranging in power from the [**Apple**](http://www.apple.com/) Macintosh to the [**Cray**](http://www.cray.com/) supercomputers. The largest measure of C's success seems to be based on purely practical considerations:

1. the portability of the compiler;
2. the standard library concept;
3. a powerful and varied repertoire of operators;
4. an elegant syntax;
5. ready access to the hardware when needed;
6. and the ease with which applications can be optimised by hand-coding isolated procedures

C is often called a "Middle Level" programming language. This is not a reflection on its lack of programming power but more a reflection on its capability to access the system's low level functions. Most high-level languages (e.g. Fortran) provides everything the programmer might want to do already built into the language. A low level language (e.g. [**assembler**](http://www.le.ac.uk/cc/glossary/ccgla.html#36)) provides nothing other than access to the machines basic instruction set. A middle level language, such as C, probably doesn't supply all the constructs found in high-languages - but it provides you with all the building blocks that you will need to produce the results you want!

FLOWCHART

Homepage

Modify

Add

delete

Search

display

EXIT

**PROGRAM OF PHONEBOOK APPLICATION USING 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];

char mother\_name[30];

long int mble\_no;

char sex[8];

Char mail [100];

Char citision\_no [20];

};

Void menu ();

Void got ();

Void start ();

Void back();

Void addrecord();

Void listrecord();

Void modifyrecord();

void deleterecord();

Void searchrecord();

int main()

{

system("color 5f");

start();

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.Exit \n\t4.Modify \t5.Search\t6.Delete");

switch(getch())

{

case '1':addrecord();

break;

case '2': listrecord();

break;

case '3': exit(0);

break;

case '4': modifyrecord();

break;

case '5': searchrecord();

break;

case '6': deleterecord();

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 mother name: ");

got(p.mother\_name);

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

scanf("%ld",&p.mble\_no);

printf("Enter sex:");

got(p.sex);

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

got(p.mail);

printf("\nEnter citizen no:");

got(p.citision\_no);

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

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 :");

exit(1);

}

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

{

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

printf("\nName=%s\nAdress=%s\nFather name=%s\nMother name=%s\nMobile no=%ld\nSex=%s\nE-mail=%s\nCitizen no=%s",p.name,p.address,p.father\_name,p.mother\_name,p.mble\_no,p.sex,p.mail,p.citision\_no);

getch();

system("cls");

}

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");

exit(1);

}

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\nFather name:%s\nMother name:%s\nMobile no:%ld\nsex:%s\nE-mail:%s\nCitision no:%s",p.name,p.address,p.father\_name,p.mother\_name,p.mble\_no,p.sex,p.mail,p.citision\_no);

}

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("CONTACT'S DATA NOT ADDED YET.");

}

else

{

ft=fopen("temp","wb+");

if(ft==NULL)

printf("file opaning error");

else

{

printf("Enter CONTACT'S NAME:");

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("NO CONACT'S RECORD TO DELETE.");

remove("temp.txt");

}

else

{

remove("project");

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

printf("RECORD DELETED SUCCESSFULLY.");

}

}

}

printf("\n Enter any key");

getch();

system("cls");

menu();

}

void modifyrecord()

{

int c;

FILE \*f;

int flag=0;

struct person p,s;

char name[50];

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

if(f==NULL)

{

printf("CONTACT'S DATA NOT ADDED YET.");

exit(1);

}

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);

printf("\nEnter father name:");

got(s.father\_name);

printf("\nEnter mother name:");

got(s.mother\_name);

printf("\nEnter phone no:");

scanf("%ld",&s.mble\_no);

printf("\nEnter sex:");

got(s.sex);

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

got(s.mail);

printf("\nEnter citizen no\n");

got(s.citision\_no);

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();

}

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);

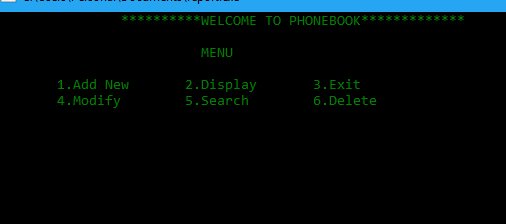
}

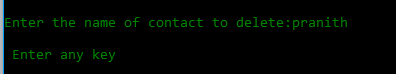
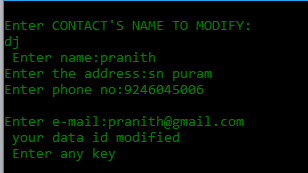
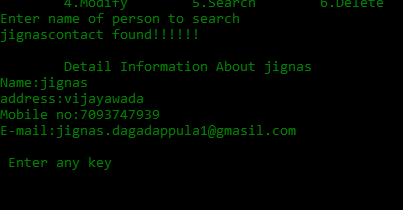
}

}while(c!=13);

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

}



****

****

**ADVANTAGES AND DISADVANTAGES**

**CONCLUSION**

The application software has been implemented successfully by using test cases .And the language used is C language. This application is used to search, delete, modify and some functions which is used to remember our friends details more easily.

**REFERENCES**

• N.B. Venkateswarlu and E.V.Prasad, C and data structures, S.Chand publishing,1st Edition,2010.

• Reema Thareja, Programming in C, Oxford University Press, 2nd Edition,2015.

• Stephen G.Kochan, Programming in C, Pearson Education, 3rd Edition, 2005.

• PradeepDey, Manas Ghosh,Programming in C, Oxford University Press, 2nd edition,2011.

• E Balagurusamy, Computer Programming,McGraw Hill Education, 1st Edition.