**PROJECT TITLE**:Phonebook Application

Adding new records into the memory, listing them, modifying them and updating, search for contacts saved, and deleting the phonebook contacts 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, gender, Father’s name, Mother’s name, phone number, email and address are asked while adding a record into the Phonebook. After the initial addition of the contacts these records are then be modified, listed, searched for and removed if they are requires a change.

The program uses many functions in. These functions are easy to understand as their name only signifies their respective operations. These functions make the program to divide into individual modules and hence it makes the program simpler to understand.

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.

**ABSTRACT**

The Title of the Program is to develop a program which deals with the combination of structures, arrays, File pointers and other functions. This program could do some operations on arrays such as insertion, deletion, sorting, searching, update, retrieve, merging, append,exit.

By implementing this program we can execute the inserted contact data, deletion of the data, searching, updating, append, exit with numbers by using arrays and file pointers. This program is implemented for only numbers that can enter into an array. To do this analysis manually it takes a lot of time and patience but by implementing this program using a high level language like C it becomes much easier. But before going to make final solution for the problem, the problem must be analysed.

First of all the basic information regarding the program which consists of complex numbers. This program is solved by using several methods like one can solve this program using user defined functions concept, loops conditions, go to statements. In this abstract we used the concept of functions, while loop, for loop, switch case and if condition’s which helps to execute the problem much easier .The following steps are followed while implementing the given program using if and while loop.

* The input is entered i.e., the value of choice (the menu no) select the particular menu.
* Next it goes to particular menu and then goto the particular function.
* It prints the resultant value which came from the execution.

The outcome of the work is one can get the required changes like inserting or deleting or sorting or merging or append or retrieve or update or exit for a given array.

**INDEX**

**S.NO TITLE PAGE NO**

1. Introduction 6
2. Aim of the Project 8

2.1 Advantages & Disadvantages

2.2 Future Implementation

1. Software & Hardware Details 8
2. Data Flow Diagram 9
3. Algorithm 13
4. Implementation 24

**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 plat forms where there will be implemented. The c language comes from the ALGOL which gives the concept structured programming to the computer science community. It was introduced early in 1960’s.

After, MARTIN RICHARDS DEVELOPED a language known as BCPL in 1967 for this in 1970’s ken Thompson created a language from BCPL and he called as “B” both BCPL and B are types less system programming languages. After finding ALGOL BCPL, AND B then from this c is evolved from that at BELL LABORATORIES in 1972 by “DENIS RITCHE”. 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 high way.C-language is robust language because c-supports richest of operators and burden functions this consist of many operators, operands, key words, special characters, many characters.

**Features of c-programming:-**

* It is a structural programming language with fundamental flow control construction.
* It is highly portable. The program written on one computer can run on another computer also without any modification or with a slight modification.
* It contains 32 keywords.
* It is simple and versatile programming language.
* It is richest than all programs.
* Dynamic memory allocation is possible in ‘c’.

**Structures**

We have seen that arrays can be used to represent a group of data items that belongs to same data type. If we want to represent a collection of data items of different data types using a single name, then we cannot use an array. C supports a constructed data type known as Structure, which is a method for packing data of different data types. A structure is a convenient tool for handling a group of logically related data items. Structures help to organize complex data in a more meaningful way. It is a powerful concept that we may often need to Use in our program design.

**Definition:**

A group of data items that belongs to different data types is known as Structure.

**‘Struct’ :** It is a keyword and is used to declare a Structure.

**Declaration of structure:**

struct struct\_name

{

Data item-1;

Data item-2;

…………

…………

Data item-n;

};

**Declaration of structure variable:**

struct struct\_name identifier;

(or)

struct struct\_name dentifier-1,identifier-2,.......,identifier-n;

**(Access operator):**

It is used to access the data items of a structure with the help of structure variable.

**Syntax:**

struct\_variable . Data item;

**AIM:**

* To develop a “Phonebook contact” application using c programming.
* This program is very useful now-a-days to store complete information under single contact number.
* This program also has options to deletion and modification of the entered contact number.

**Advantages:**

* It becomes easy for the user to store complete information (e-mail id, address, e.t.c) about his contact.
* It is easy for the user to just search his required contact number by just typing name of the contact.

**Disadvantages:**

* Sometimes it becomes difficult to store more contacts (over 150).

**Future Enhancements:-**

* It becomes even difficult to store contacts with two or more contact numbers

**SOFTWARE REQUIREMENTS:**

* This application is developed in Microsoft windows xp or later operating system.
* This Phonebook application is coded and made using the following compilers:
  1. Code::blocks.
  2. Turbo c.

**HARDWARE REQUIREMENTS**:

* This Application size is 33Kb and the size of the code is 5Kb so such amount of memory is required from hard disk.
* RAM: minimum 256MB.
* Mouse, keyboard.

**DATA FLOW DIAGRAM:**

Menu for Phonebook Application

1

Read menu no

n

Case 1:

Case 2:

Case 3:

Case 4:

Case 5:

Case 6:

Default:

Read menu no

Print All the contacts in the Phonebook

Read menu no

Read Name,Address,Father Name,Mother Name,Phone number, Gender

Print Record saved

Read menu no

Read the name entered to modify

Read the modified details

Name,Address,Father Name,Mother Name,Phone number,Sex

Print Your data modified

Read menu no

Read the name to be searched

Print all the details of the searched contact

Read menu no

Read the name of the contact to be deleted

Print Record deleted successfully

**ALGORITHM:**

1. Start.
2. Display the options on the screen.
3. Read name, address, father name, mother name, mobile number, gender.
4. Print “WELCOME TO PHONEBOOK” and “Menu” on the screen
   1. Add contact
   2. Print all contacts
   3. Modify contacts
   4. Search contact
   5. Delete contact
   6. Close phone book
5. Now for Addrecord(),arranging data as Enter name,Enter address,Enter father name,Enter mother name,Enter phone book no, Enter Gender and finally operations record saved ,Enter any key.
6. Now for listrecord(),arranging data as
   1. if data is empty print file opening error in listing
   2. Elseprint your record: name,address,father,mother,mobile no, Gender.
   3. After all these print enter any key
7. Now for searchrecord(),arranging data as
   1. if data is insufficient print error in opening
   2. Else
   3. print enter name of contact to be searched and
      1. If (strcmp(p.name,name)==0)
      2. print detail information about name and name,address,father name,mother name,mobile num, gender.
   4. After entering all these, print enter any key
8. Now for deleterecord(),arranging the data as
   1. If (f==null)
   2. print contact’s data not added yet
      1. else
      2. if(f==null)
      3. print file opening error
      4. Else
      5. Print enter contact’s details.
9. if (flag!=1)
   1. print no contact’s record to delete
   2. Else
   3. print record deleted successfully
   4. After entering all the data print enter any key
10. for modifyrecord(),arranging the data as
    1. If (f==null)
    2. print contact’s data not added yet
    3. Else
    4. enter contact’s name to modify
       1. Print enter name , enter address , enter father name, enter mother name,mobile number, sex after entering the data
       2. print your data is modified
       3. Else
       4. print data is not found
       5. Finally print enter any key.
11. Redisplay the menu for user required input.

**Code of the application:**

#include<conio.h>

#include<string.h>

#include<stdio.h>

struct contact

{

char nam[50];

char adres[50];

long int m\_no;

char gender[10];

char m\_id[100];

};

void Menu();

void Data();

void ARec();

void ListRec();

void ModRec();

void DelRec();

void SRec();

int exit();

int main()

{

Menu();

return 0;

}

void Menu()

{

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

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

printf("\t1.List Contacts\t2.Add New Contact\t3.Modify \n");

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

switch(getch())

{

case '1': ListRec();

break;

case '2': ARec();

break;

case '3': ModRec();

break;

case '4': SRec();

break;

case '5': DelRec();

break;

case '6': exit();

break;

default: printf("\n Enter 1 to 6 only.");

getch();

}

}

int exit()

{

Menu();

return 0;

}

void ARec()

{

FILE \*f;

struct contact p;

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

printf("\n Name: ");

Data(p.nam);

printf("\nAddress: ");

Data(p.adres);

printf("\nMobile no.:");

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

printf("Gender:");

Data(p.gender);

printf("\ne-mail:");

Data(p.m\_id);

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

fflush(stdin);

printf("\nrecord saved");

fclose(f);

getch();

Menu();

}

void ListRec()

{

struct contact p;

FILE \*f;

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

if(f==NULL)

{

printf("\n No contacts detected yet!:\n");

exit();

}

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

{

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

printf("\nName=%s\nAdress=%s\nMobile no=%ld\ngender=%s\nE-mail=%s",p.nam,p.adres,p.m\_no,p.gender,p.m\_id);

getch();

}

fclose(f);

getch();

Menu();

}

void SRec()

{

struct contact p;

FILE \*f;

char name[100];

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

if(f==NULL)

{

printf("\n Error in opening\n");

exit();

}

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

Data(name);

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

{

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

{

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

printf("\nName:%s\naddress:%s\nMobile no:%ld\ngender:%s\nE-mail:%s",p.nam,p.adres,p.m\_no,p.gender,p.m\_id);

}

else

printf("Not found");

}

fclose(f);

getch();

Menu();

}

void DelRec()

{

struct contact p;

FILE \*f,\*ft;

int flag;

char name[100];

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

if(f==NULL)

printf("Contact data not yet added.");

else

{

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

if(ft==NULL)

printf("file opening error");

else

{

printf("Enter contact name:");

Data(name);

fflush(stdin);

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

{

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

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

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

flag=1;

}

fclose(f);

fclose(ft);

if(flag!=1)

{

printf("No Record To Delete.");

remove("temp.txt");

}

else

{

remove("project");

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

printf("Record Deleted");

}

}

}

getch();

Menu();

}

void ModRec()

{

int c;

FILE \*f;

int flag=0;

struct contact p,s;

char nam[50];

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

if(f==NULL)

{

printf("No Contact added yet!\n");

exit();

}

else

{

printf("Enter Name to modify:\n");

Data(nam);

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

{

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

{

printf("\nName:");

Data(s.nam);

printf("\nAddress:");

Data(s.adres);

printf("\nPhone no:");

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

printf("\nGender:");

Data(s.gender);

printf("\ne-mail:");

Data(s.m\_id);

fseek(f,-sizeof(p),SEEK\_CUR);

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

flag=1;

break;

}

fflush(stdin);

}

if(flag==1)

printf("\n Your contact is modified");

else

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

fclose(f);

}

getch();

Menu();

}

void Data(char \*Name)

{

int i=0,j;

char ch,cha;

do

{

ch=getch();

if(ch!=8&&ch!=13)

{

\*(Name+i)=ch;

putch(ch);

i++;

}

if(ch==8)

{

if(i>0)

{

i--;

}

for(j=0;j<i;j++)

{

cha=\*(Name+j);

putch(cha);

}

}

}while(ch!=13);

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

}

**Conclusion:**

This program makes the user simpler to connect to his contact. The contact personal information and family information is stored under a single number this would benefit the user to easily search and locate his required contact. This program deals with four operations of adding contacts, deleting them, modifying, searching according the user’s choice. Each operation is made as an individual function and so control enters to different structures and all the data added or modified or deleted is going to be stored in a .txt fileusing FILE pointers.