# Chapter 1

# Introduction

Earlier phonebook was a book that printed yearly and distributed to the people by phone companies. The people used phonebook to look up the contact number of people who they had met, or friends and family members whose numbers are misplaced. All contacts are listed in that phonebook. If we need to Search the contact details takes more time and if we want to add or modify the contact we need to contact the phone company and we need to add or modify the details.

But at present phonebook is digitalized. If we need to add the contact details we can add without help of company or any assistant we can add the contact of the person very easily. We can also search the contact information very easily just by typing the contact name or contact number. Updating the details of a contact can also be performed easily. The same is implemented by using c++ language.

* 1. **Motivation of Study**

Developing the phonebook using c language in order to store the contact name and contact number. The basic functions that we can perform in this project are:

* Adding the records
* Listing the records
* Searching the records by name
* Searching the records by phone number
* Modifying the records

**1.2 Problem Statement**

If the person is using the manual system to store the contact details of his family, friends and others .it is very difficult to make a record of their details and it is difficult to find the details whenever he is needed and also it takes more time to make a record and to search the records.

So by using phonebook we can store the records in easy manner and we can obtain the details easily.

**Chapter 2**

# System requirements:

**2.1 Hardware System Configuration:**

Processor - Intel Core i5

Speed - 1.8 GHz

RAM - 256 MB (min)

Hard Disk - 10 GB

**2.2 Software System Configuration:**

Operating System - Fedora

Programming Language - C++

Compiler - C Compiler

**Chapter 3**

**System Design**

**3.1 System Modules:**

This project consists of following modules:

Module – 1: Header Files

Module – 2: Declaring Class

Module – 3: Get data ( )

Module – 4: Display ( )

Module – 5: Main ( )

Module – 6: Adding records ( )

Module – 7: Listing records ( )

Module – 8: searching records ( )

**3.2 Architecture:**

LOGIN

False

Enter username & password

**True**

Adding record

Searching records by phone no

Modifying record

Searching record by name

Listing records

Enter name

Enter name

Enter name

Enter phone number

Enter phone no

Enter phone no

Display records

Display records

Displaying records

**3.3 Algorithm:**

Step 1: start

Step 2: Selecting the operation

1. Adding new record
2. Displaying the record
3. Search by name
4. Search by phone number
5. Update the information
6. Exit

Case1:

1.1 Accepting the name and phone number

1.2 Writing the information into the file

Case 2:

2.1 Reading the content from the file

2.2 Displaying all records

Case 3:

3.1 Accepting the name of the person to search

3.2 Displaying the record

Case 4:

4.1 Accepting the phone number to search

4.2 Displaying the record

Case 5:

5.1 Accepting the name of the person

5.2 If name is found

5.2.1 Entering the new phone number

5.2.2 Replacing the phone number

5.3 If name is not found

5.3.1 Displaying record not found

Step 3: Closing the file

Step 4: Stop

**3.4 Code and Implementation:**

#include <iostream.h>//basic input and output services

#include <fstream.h>//acess the file

#include <string.h>//macro definations

#include <iomanip.h>

#include <conio.h>

class phoneBook{

char name[20],phno[15];

public:

void getdata();//input function

void showdata();//display function

char \*getname(){ return name; }

char \*getphno(){ return phno; }

void update(char \*nm,char \*telno){

strcpy(name,nm);

strcpy(phno,telno);

}

};

void phoneBook :: getdata(){

cout<<"\nEnter Name : ";

cin>>name;

cout<<"Enter Phone No. : ";

cin>>phno;

}

void phoneBook :: showdata(){

cout<<"\n";

cout<<setw(20)<<name;

cout<<setw(15)<<phno;

} void main(){

clrscr();

int u,p;

cout<<"\t\t\tenter the username:"<<endl;

cout<<"\t\t\t";

cin>>u;

cout<<"\n\n";

cout<<"\t\t\tenter the password:"<<endl;

cout<<"\t\t\t";

cin>>p;

if(u==12345 && p==12345)

{

phoneBook rec;

fstream file;//helps to read or write a file

file.open("phone.txt", ios::ate | ios::in | ios::out | ios::binary);

/\*ate=add data at the end of the file,in=to input operations,out=to output operations,to open in binary mode\*/

char ch,nm[20],telno[6];

int choice,found=0;

while(1){

clrscr();

cout<<"\n\*\*\*\*\*Phone Book\*\*\*\*\*\n";

cout<<"1) Add New Record\n";

cout<<"2) Display All Records\n";

cout<<"3) Search Telephone No.\n";

cout<<"4) Search Person Name\n";

cout<<"5) Update Telephone No.\n";

cout<<"6) Exit\n";

cout<<"Choose your choice : ";

cin>>choice;

switch(choice){

case 1 : //New Record

rec.getdata();

cin.get(ch);

file.write((char \*) &rec, sizeof(rec));

break;

case 2 : //Display All Records

file.seekg(0,ios::beg);/\*helps to find the function location of the key word\*/

cout<<"\n\nRecords in Phone Book\n";

while(file){

file.read((char \*) &rec, sizeof(rec));/\*reading the file data\*/

if(!file.eof())//if it is not end of file

rec.showdata();

}

file.clear();//clears the dat ainside the text file

getch();

break;

case 3 : //Search Tel. no. when person name is known.

cout<<"\n\nEnter Name : ";

cin>>nm;

file.seekg(0,ios::beg);

found=0;

while(file.read((char \*) &rec, sizeof(rec)))

{

if(strcmp(nm,rec.getname())==0)

/\*comparing the entered name and the names in the text file\*/

{

found=1;

rec.showdata();

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

getch();

break;

case 4 : //Search name on basis of tel. no

cout<<"\n\nEnter Telephone No : ";

cin>>telno;

file.seekg(0,ios::beg);

found=0;

while(file.read((char \*) &rec, sizeof(rec)))

{

if(strcmp(telno,rec.getphno())==0)

/\*comparing the entered phone no and the numbers in the text file\*/

{

found=1;

rec.showdata();

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

getch();

break;

case 5 : //Update Telephone No.

cout<<"\n\nEnter Name : ";

cin>>nm;

file.seekg(0,ios::beg);

found=0;

int cnt=0;

while(file.read((char \*) &rec, sizeof(rec)))

{

cnt++;

if(strcmp(nm,rec.getname())==0)

{

found=1;

break;

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

else

{

int location = (cnt-1) \* sizeof(rec);

cin.get(ch);

if(file.eof())

file.clear();

cout<<"Enter New Telephone No : ";

cin>>telno;

file.seekp(location);

rec.update(nm,telno);

file.write((char \*) &rec, sizeof(rec));

file.flush();

}

break;

case 6 : goto out;

}

}

out:

file.close();

}

else

{

cout<<"\t\tusername and password missatch!!"<<endl<<"\t\t\ttry angain";

getch();

}

}

**3.4.1 Output (snap-shots)**

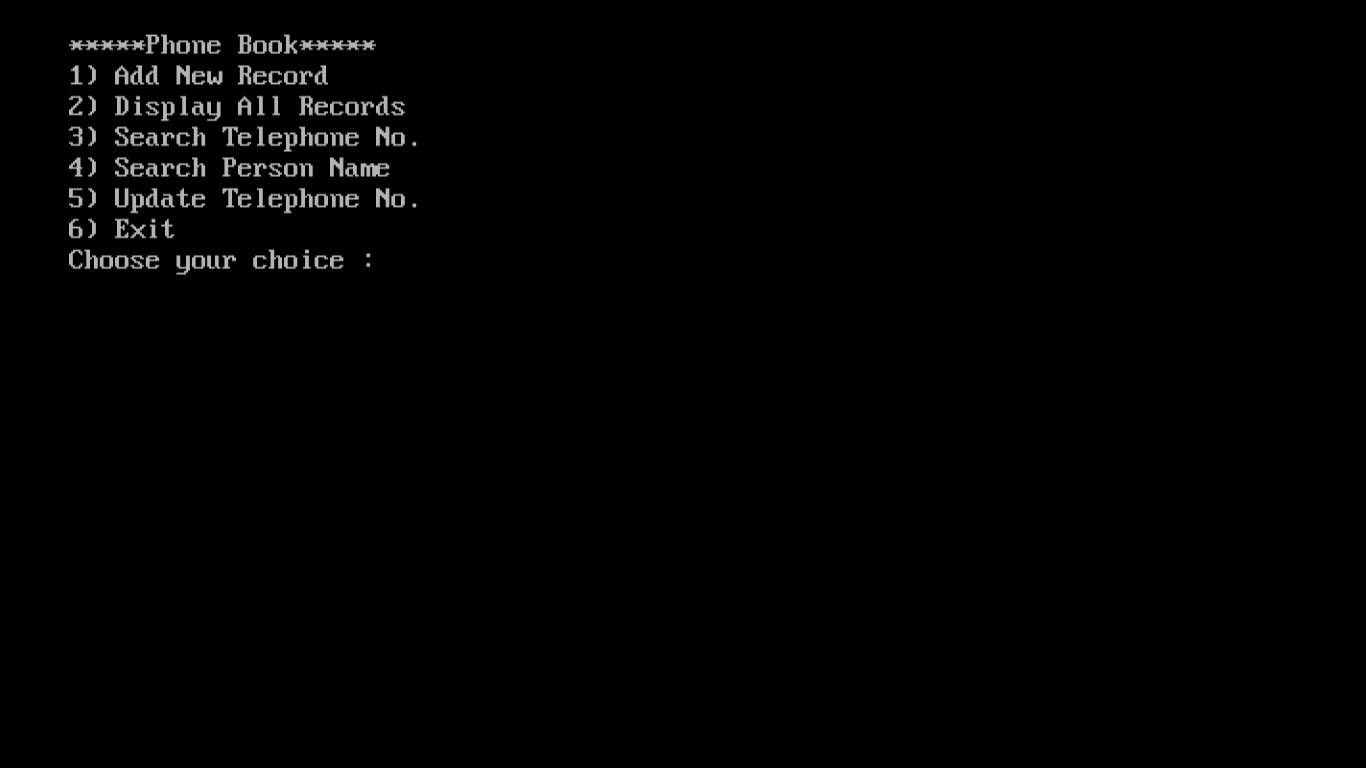
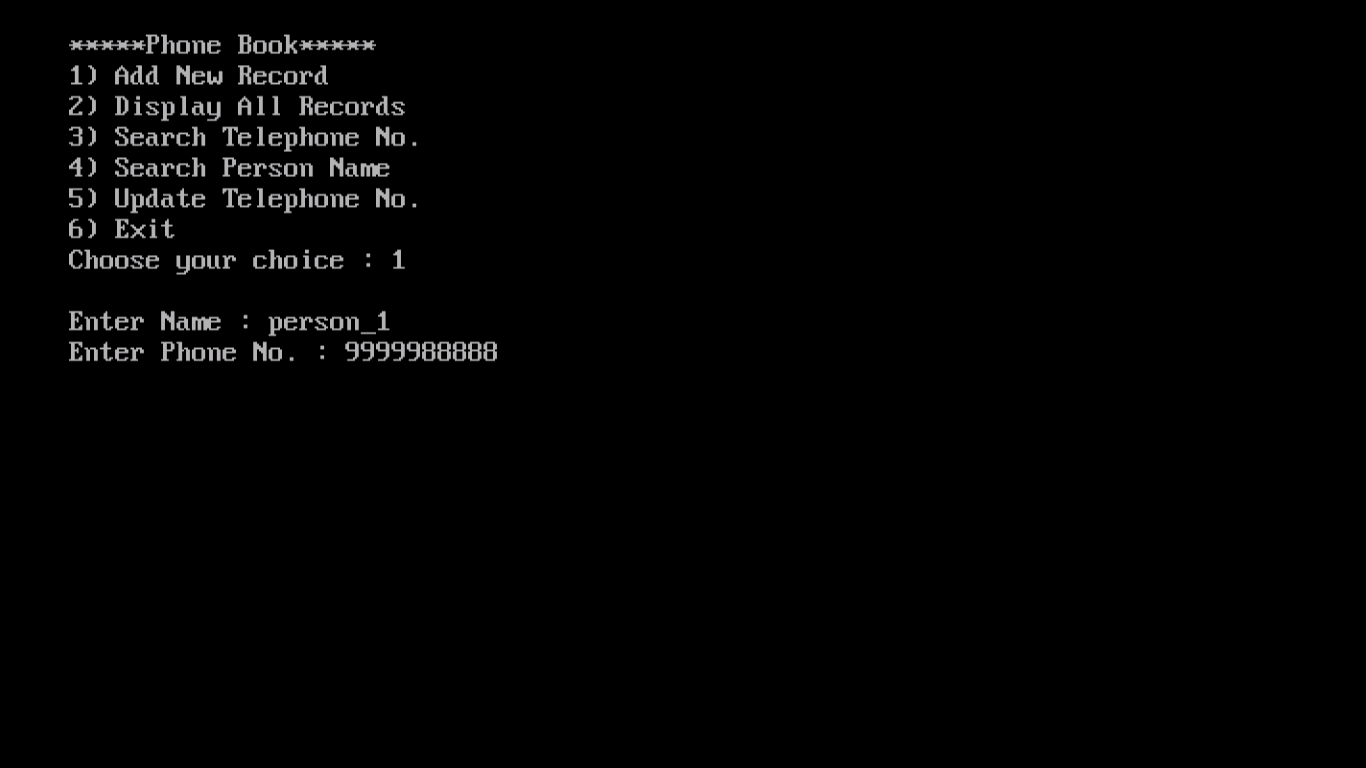


Fig.1 displaying the menu.

After debugging of the code we will displayed a menu of the operation that can be performed in this phonebook project. And we need to enter the option number by selecting the operation to be performed from the menu.

Fig.2 entering the option and storing the contacts

If we select the option 1 i.e. adding the record, we will be asked the contact details and we need to enter the contact name and contact number and press enter. Then it will store the contact with the given contact name and contact number.

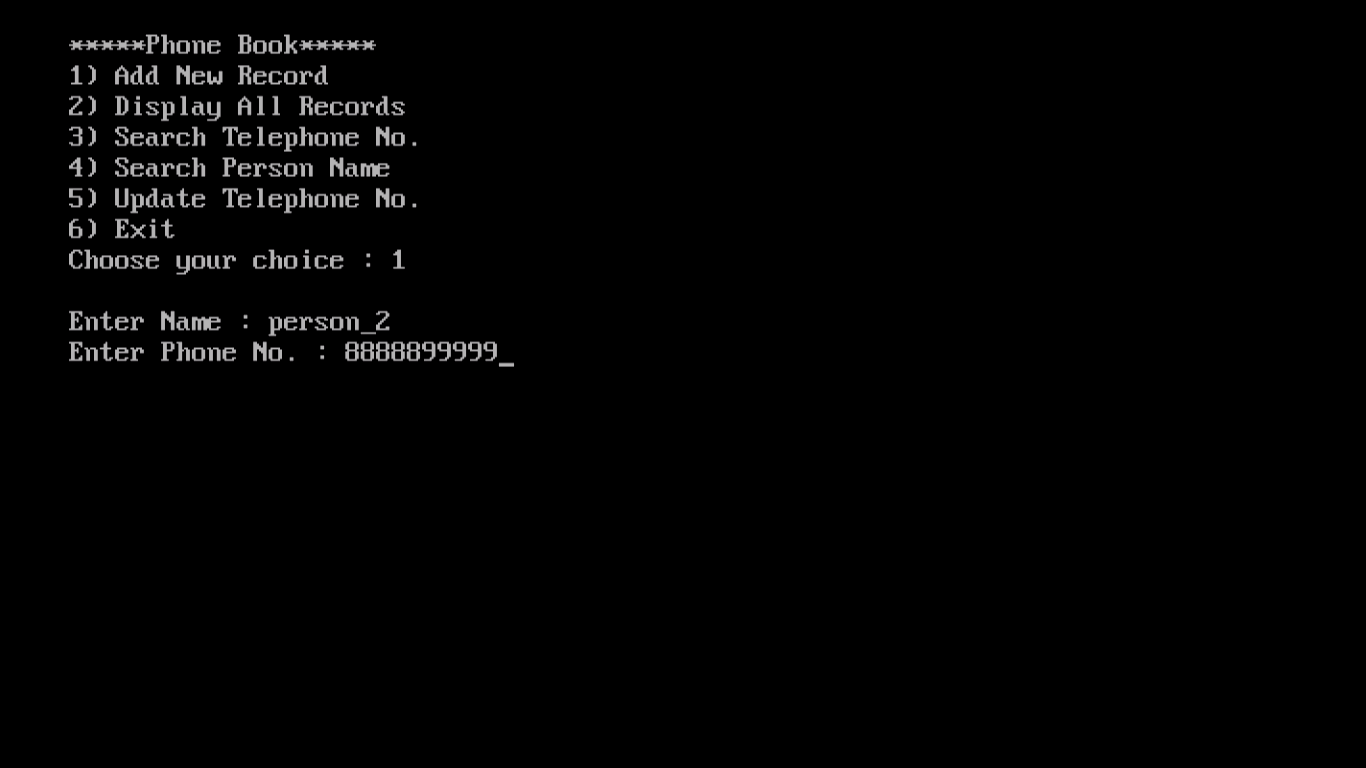


Fig.3 storing the 2ndcontact.

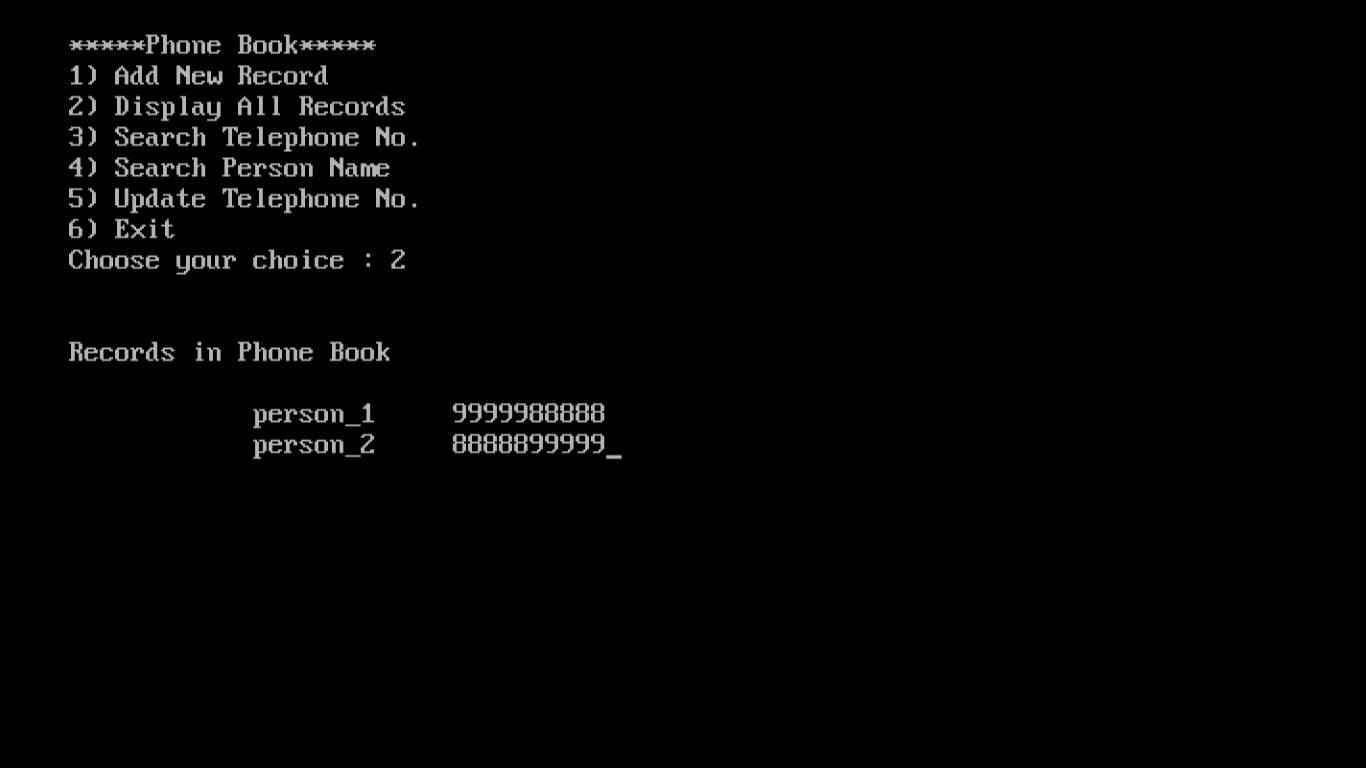
We can store the multiple contacts by selecting the option 1 multiple times and giving the details.

Fig 4:Displaying all records

Displaying all records is the option available in the menu. This will help us to view all the records which are already stored in the phone book. In order to display all contacts we need no select the option 2 i.e. display all records. By entering the option as 2 and the records will be displayed.

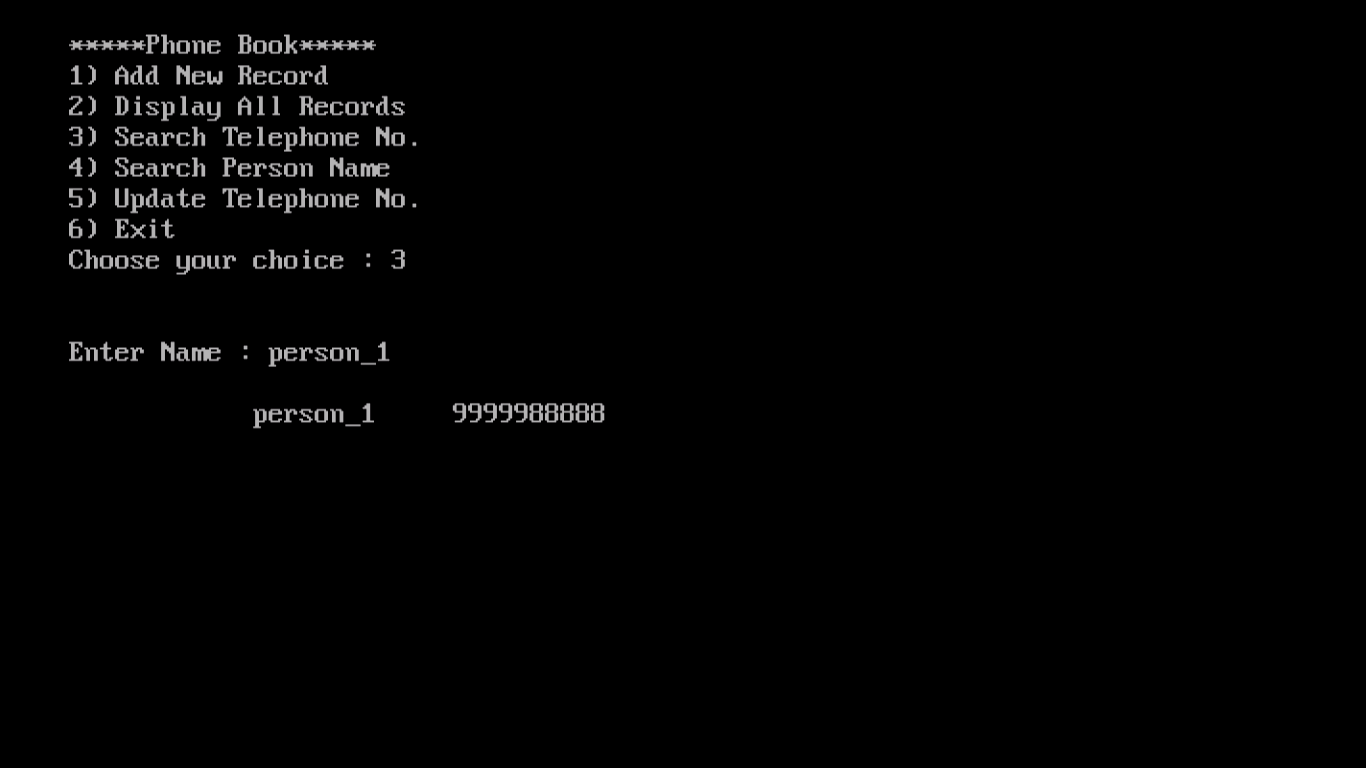


Fig.5 searching the contact by name

Searching the contact number by contact name is a special operation available in this project which will enable us to search for any contact which is already stored in the phonebook.

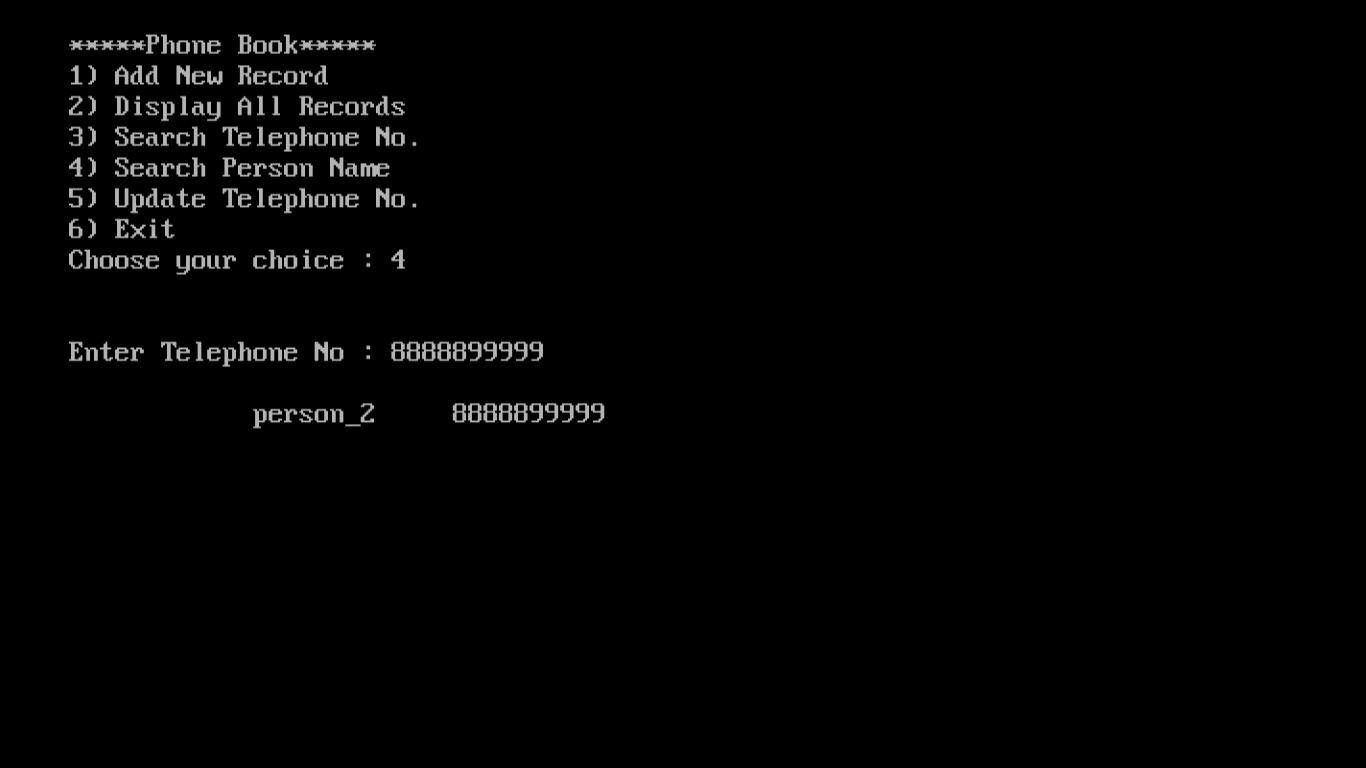
In order to search the contact number with contact name we need to select the 3rd option from the menu and we will be asked contact name. After entering the contact name the details will be displayed.

Fig 6. Searching contact by contact number.

Searching the contact by contact number is a special operation available in this project which will enable us to search for any contact which is already stored in the phonebook.

In order to search the contact number with contact name we need to select the 4th option from the menu and we will be asked contact name. After entering the contact number the details will be displayed.

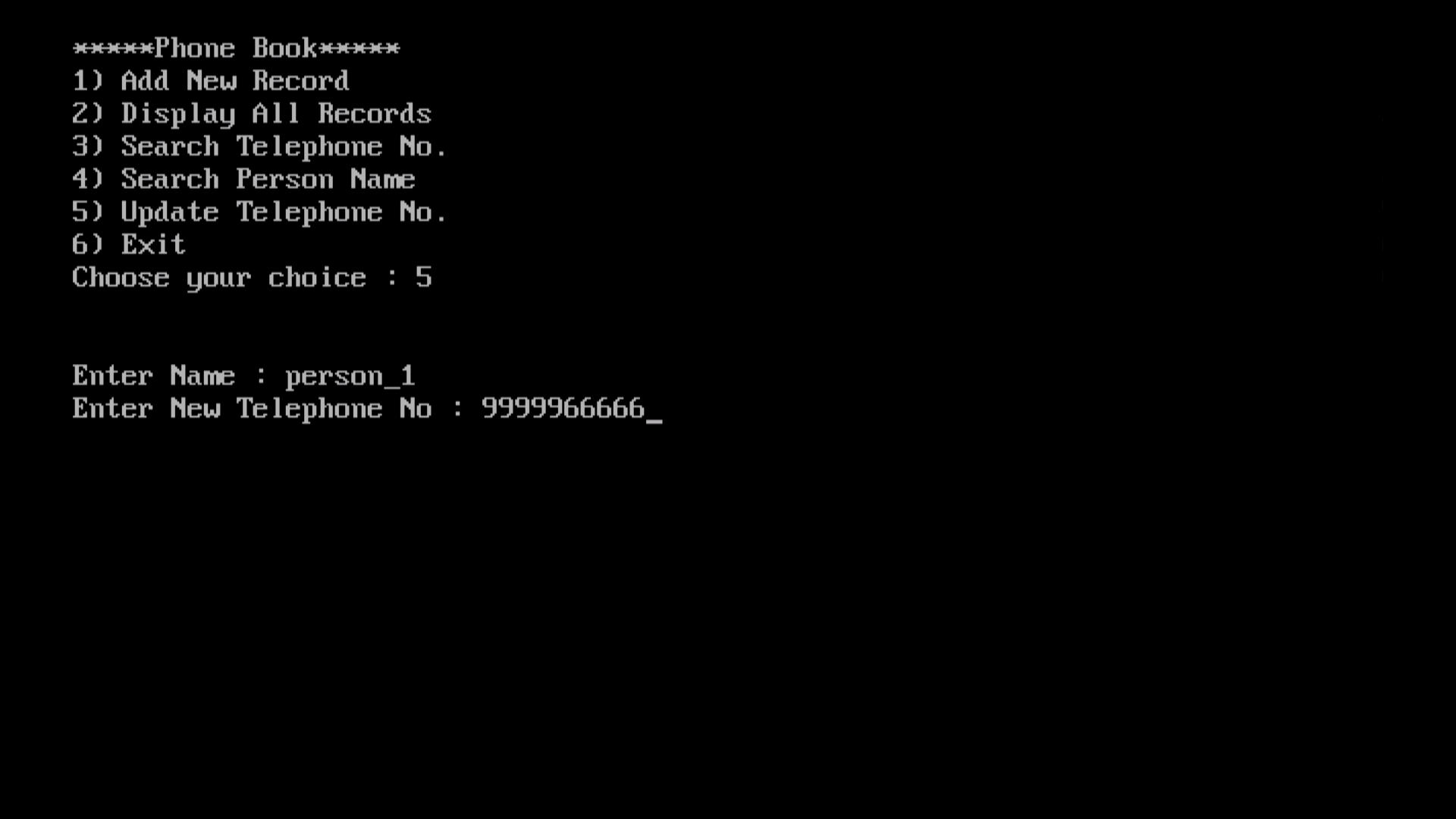


Fig 7. Updating the contact number

Updating the contact number is the option available in phonebook used in order to update the contact number. In order to perform this operation we need to select the option 5 i.e. update contact number.

In this will be asked the contact name, after entering the contact name we should enter the new contact number and press enter. Finally the new contact number will be updated in the old contact only.

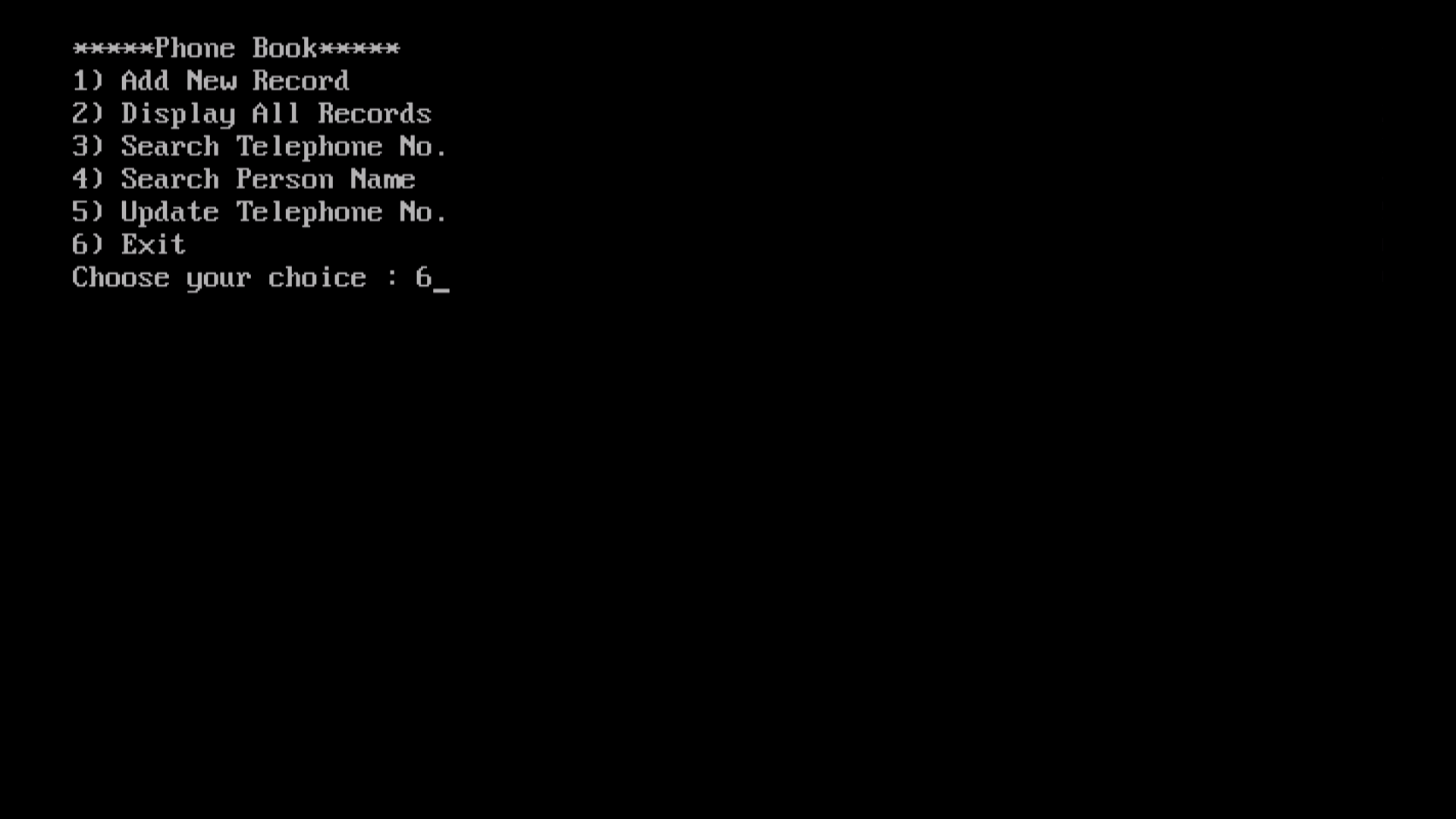


Fig 8. Exit

In order to exit from this operation we need to enter the option 6.

This will exit from the menu and close the program.

**Chapter 4**

**Results and Discussion**

Phonebook expected to be well managing contact records. It will be easier to the user to add the record, list the records, search the records, modifying the records. This makes user easier to save the contacts and retrieve the saved records whenever he is required.

**Conclusion**

This project is being developed keeping in view the entire requirements that are being specified. This project fulfills all the requirements for single user to Add, search and make modification in his Phonebook and it will definitely enhance the working efficiency of the previous system. The various functionalities that are provided by this project will definitely help the user of the existing system to enhance the management of the contacts and keeping the contact details of many/every person whom he knows with more accurately and more efficiently.

**References**

The following books were very helpful during the completion of project:

1. **The C Programming Language 2nd Edition**

* **by Brian W. Kernighan and Dennis M. Ritchie in** April 1988

1. **The C++ Programming Language (4th Edition)**

* **By** Bjarne Stroustrup created C++ in 1979.

1. **C++ Primer (5th Edition)**

* **by Stanley B. Lippman in 2012**

1. **Effective Modern C++ (1st Edition)**

* **by Scott Meyers in 2014**

1. **Programming in C (4th Edition)**

* **by Al Kelley and Ira Pohl in 2014**

# Websites:

# [www.dataoncpp.com](http://www.dataoncpp.com)

# [www.codewithcpp.com](http://www.codewithcpp.com)

# <https://code-reference.com/c>++

1. <https://stackoverflow.com/questions/41696230/haskell-phonebook-adding-elements>
2. <https://www.tutorialspoint.com/cplusplus>
3. <https://codereview.stackexchange.com/questions/181495/phone-book-program-in-c>++