ASSIGNMENT

MAY 2022 SEMESTER

SUBJECT CODE	:	**CCP203**
SUBJECT TITLE	:	**COMPUTER PROGRAMMING**
LEVEL	:	**BACHELOR**

STUDENT'S NAME	:	**SANJINA RAI**
MATRIC NO.	:	**C30105220028**
PROGRAMME	:	**BICT (HONS)**
ACADEMIC FACILITATOR	:	**JEEWAN RAI**
LEARNING CENTRE	:	**VIRINCHI, NEPAL**

INSTRUCTIONS TO STUDENTS

This assignment consists of **THREE (3)** questions. Answer the questions.

Plagiarism in all forms is forbidden. Students who submit plagiarised assignment will be penalised.

Your assignment will be examined based on the followings
a complete working solution.
ability of using methods available in the learning materials.

4) This assignment carries a **60%** weightage toward final grade.

THERE ARE THREE [3] PAGES OF QUESTIONS, INCLUDING THIS PAGE.

FINAL PROJECT

SEMESTER I

PHONEBOOK MANAGEMENT SYSTEM

BY

SANJINA RAI

BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGY

SCHOOL OF SCIENCE AND TECHNOLOGY

ASIA e UNIVERSITY

This report is prepared to fulfill the requirement of

CCP203 Final Project

By :

SANJINA RAI

C30105220028

SCHOOL OF SCIENCE AND TECHNOLOGY
ASIA e UNIVERSITY

SEMESTER 1st

DISCLAIMER

CCP203 – FINAL PROJECT

I am accountable for the accuracy of every viewpoint, expert remark, factual report, data, figure, illustration, and picture featured in this report. It is entirely my duty to ensure that the report I have provided has been checked for copyright or other ownership restrictions. Asia e University disclaims all responsibility for the veracity of any opinion, report, and other technical or factual data, as well as any claims of ownership or copyright.

SANJINA RAI
C30105220028

ACKNOWLEDGEMENT

Firstly, I would like to express my heartfelt appreciation to Mr. Roshan KC, the coordinator of Virinchi College for approving this project. Mr. Animesh Regmi deserves a lot of credit for his kind advice and support during the difficult times. His helpful supervision, invaluable assistance and motivation were a significant driving force for the successful completion of my project.

Furthermore, I would like to thank all my friends and families for their help, cooperation and encouragement in completing this project on time. Completing this project without them would have been a difficult task for me so, I am very happy and grateful for their love and support.

Last but not least, I am extremely grateful for all the supports and help I have received from the seniors from my college. They were kind and patient when they guided and assisted me for any queries or confusion I had regarding the project.

ABSTRACT

“Phonebook” is a simple application that you can find on your mobile as a contact app. It is especially generated and designed for the users for file handling operations such as how to add, modify, list and delete phone records using file. There are more than 600 million mobile phone users around the world and its success lies in being available anytime and anywhere.

Phonebook Management System helps us to access our contact anywhere and everywhere whether at work or at home. A phonebook application's primary goal is to quickly locate a person's or organization's address. We can, for example, look up a person by their name, address, email, phone number, or location. This system can be used to monitor daily activities by schools, bus companies, government agencies, offices, and businesses.

Table of Contents

1. INTRODUCTION

This whole project is designed in ‘C’ language which allows user to perform some basic phonebook operations like in mobile. The basic functions of this projects include listing the record, adding new record, searching person, updating or deleting contacts which make up the main menu in this application. Personal information such as name, country code, phone number, sex, email is asked while adding a record into the phonebook. These records can be searched, removed, updated or deleted. This program creates an external file to store the user’s data permanently and performs the file handling operations. There are more than 600 million mobile phone users in the world and remembering all the contacts can be a difficult thing. People can store contact information and use it anywhere they want. If they need to know any person’s detail, they will easily get the information they are looking for through a number search. It will enable people to easily locate person’s contact so it also saves time.

2. MOTIVATION

The main reason in using this project is the increasing use of the mobile phone. This is the age of technology where communicating from one place to another from any corner of the world is possible. Since we are human and not robot, it is not possible to remember every contact details. This application can help us to store the details about the person which will make it easier for any user to use it anywhere and everywhere. Since, people are getting busier and busier in their daily life, this application is a good choice for anyone to record details and information.

3. OBJECTIVE

The main aim of this project is:

- To develop an “Phonebook contact” application using C programming.
- To store complete information under a single contact number.
- To delete and modify the entered contact number.

4. LITERATURE REVIEW

The Phone Book system is a small web application created for this project. We used to keep all of our important contact information in books and papers. In this section, we proposed a new system in which we can store all of the details in a central repository. If we forget the information book in the manual method, it is extremely difficult to obtain contact information. We can see our contacts from anywhere in the world by using this application. The login page is secure, and no one can see our contacts without proper authentication. In this project, we can save our contacts and addresses, search for them by name, and view them all at once.

5. PROBLEM STATEMENT

The following are a few project limitations: -

Contacts are given names. Strings are used to represent names (for example, "Sanz"). A name must be recorded for each contact (for example, contact 3 has the name "Sanz"). Each contact has only one name (for example, contact 3 cannot have the names "Sanz" and "Rosh"). It is possible that more than one contact has the same name (for example, contact with id 2 may be called "Sanz" and contact with id 3 may also be called "Sanz"). Each contact-name combination is unique (for example, the fact that contact 3 has the name "Sanz" cannot be recorded more than once). Hence, sometimes it becomes difficult to store more contacts.

Phone numbers are assigned to contacts. Integers (for example, 99224434) are used to represent phone numbers. Each contact may have one or more phone numbers (e.g., mobile or landline). It is possible that a phone number is shared by more than one contact. Every contact-phone number combination is distinct. So, it becomes even difficult to store contacts with two or more contact numbers.

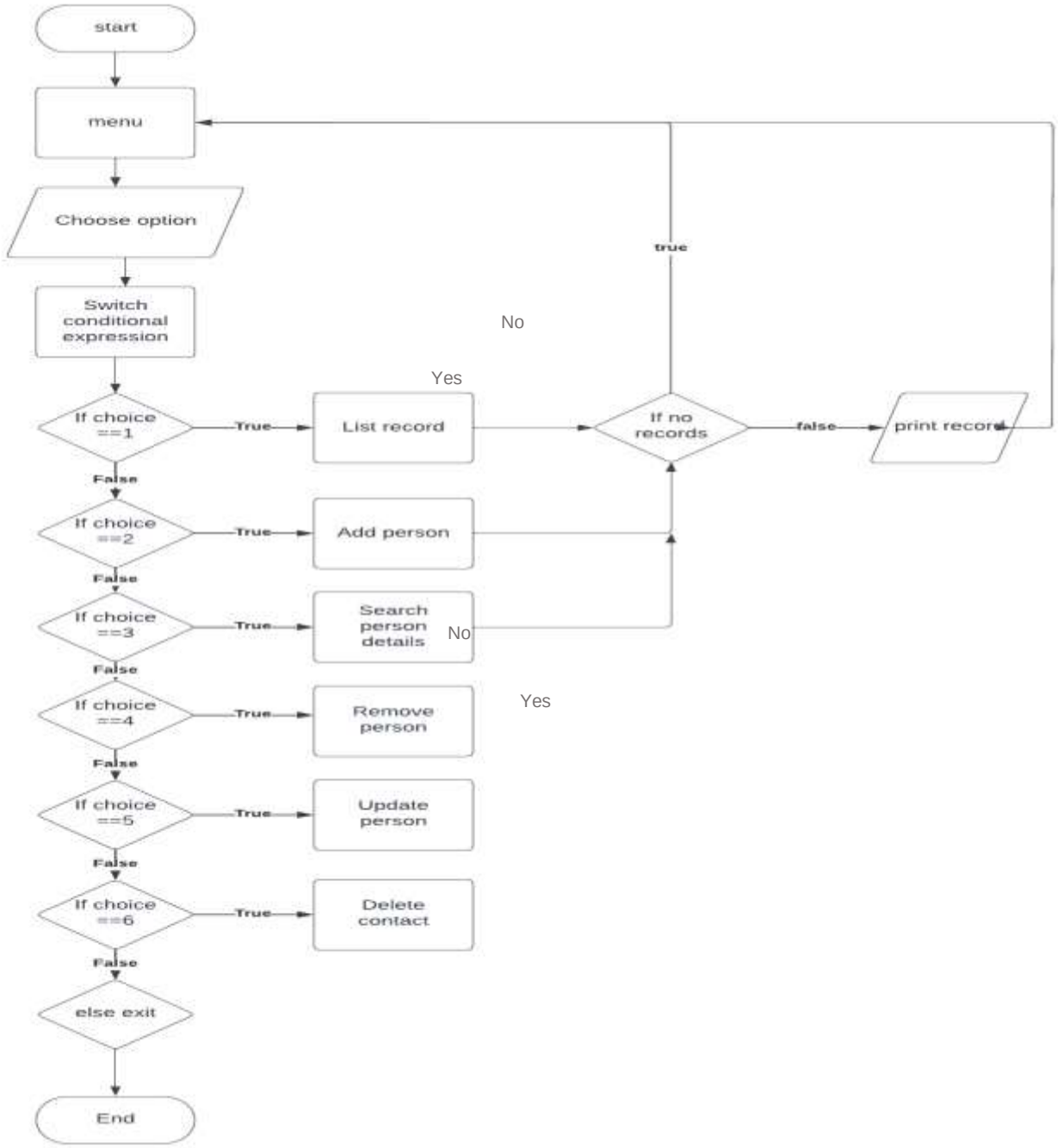
6. **ALGORITHM**

The algorithm of the entire system is given below:

Steps	**Algorithm**
Step 1	Start
Step 2	Welcome To My Phone Book
Step 3	Choose option
Step 4	If choose=1 then go to list record
Step 5	Else if choose=2 then go to add person
Step 6	Else if choose=3 then go to search person detail
Step 7	Else if choose=4 then go to remove person
Step 8	Else if choose=5 then go to update person
Step 9	Else if choose=6 then go to delete all content
Step 10	Else go to exit phonebook
Step 11	End

7. **FLOWCHART**

A general pictorical representation of the program is given below in the flowchart:



Fig(a) Flowchart

8. IMPLEMENTATION

We execute our program with the help of string and i/o libraries. To implement the goals, following methodology are used.

1. This system begins with a login authentication in which strings from the file where data is stored are compared.

2. It stores, reads, and writes data in files using the file handling concept.

9. FUTURE ENHANCEMENTS

The project “Phonebook Management System” is an application for performing simple phonebook operations like in your mobile. It is what you need for storing phone records. This system is certain to solve the problem of statements that have been explained in this report. The information stored help the user to locate the person or organization contact address. Some disadvantages are already highlighted in this report, so future improvements will focus on the limitation.

10. OUTPUT SCREEN

The output screen of some features is provided below:

Fig: Login page

Fig: Main menu

Fig: Add person

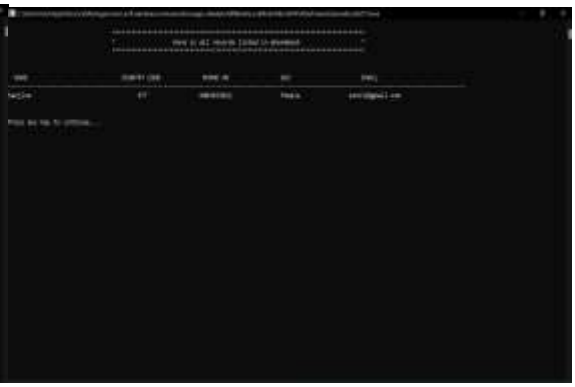


Fig: List record

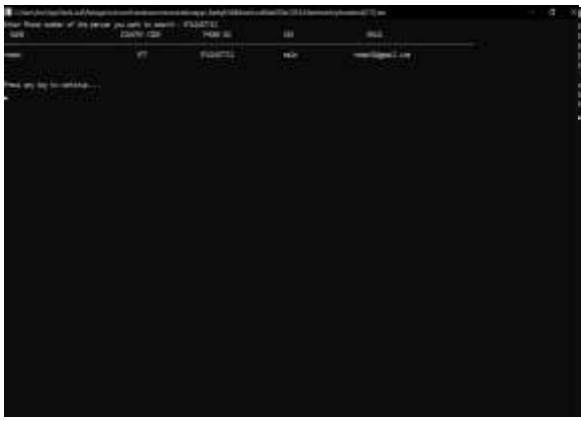


Fig: Search person

Fig: Update person

Fig: Remove person

Fig: Delete contact

Fig: Exit Phonebook

11. REFERENCES

1. Admin. (n.d.). Retrieved from MalikProgrammers: https://malikprogrammers.com/2021/08/23/simple-mini-project-phonebook-in-c/
2. ahme144576. (n.d.). Retrieved from Coursehero: https://www.coursehero.com/file/44015077/PHONEBOOK2pdf/
3. Suarez, A. J. (n.d.). Retrieved from ITSourceCode: https://itsourcecode.com/free-projects/c-projects/phone-book-management-system-in-c-with-source-code/
4. Evangelista, A. (2020). *Itsourcecode*. Retrieved from https://itsourcecode.com/free-projects/c-projects/
5. KAZI-SHAMIM-SHAHAREAR. (2018). *Github*. Retrieved from https://github.com/1Shaharear/PhoneBook-Application-using-C-programming/blob/master/phonebook%20application.c

12. PROJECT CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define ENTER 13
#define TAB 9
#define BCKSPC 8

struct person
{
    char name[30];
    char country_code[4];
    char mble_no[20];
    char sex[8];
    char mail[100];
};

// Defining person data type.
typedef struct person person;

// All function declaration.
void remove_all();
void print_menu();
void add_person();
void list_record();
void search_person();
void remove_person();
void update_person();
void take_input(person *p);

char takepassword(char pwd[20]){
     int i;
     char ch;
     while(1){
          ch = getch();
          if(ch == ENTER || ch == TAB){
```

```
            pwd[i] = '\0';
            break;
        }
        else if(ch == BCKSPC){
            if(i>0){
                i--;
                printf("\b \b");
            }
        }
        else{
            pwd[i++] = ch;
            printf("* \b");

        }
    }
}

// Program starts here.
int main()
{
    int w=0;
    char username[20]="sanjina",uname[20];
    char password[20]="raee",pas[20];
    top:
    printf("\t\t\t\t
    ====================================\n");
    printf("\t\t\t\t    ||------------Login Page-----------
||\n");
    printf("\t\t\t\t
    ====================================\n");
    printf("\nEnter username\t");
    scanf("%s",&uname);
    printf("\nEnter password\t");
    takepassword(pas);
    if(strcmp(uname,username)==0&&strcmp(pas,password)==0){
        printf("\n\nlogin success");
```

```
            getch();

        }
        else{
            printf("\n\nlogin is not sucessful try again\n");

            getch();
            while(w<2){
                w++;
                system("cls");
                goto top;
            }
            exit(0);

        }
    start();

    return 0;
}

// This function will start our program.
int start()
{
    int choice;
    while(1)
    {
        print_menu();
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                list_record();
                getchar();
                getchar();
                break;
            case 2:
```

```
                add_person();
                getchar();
                getchar();
                break;
            case 3:
                search_person();
                getchar();
                getchar();
                break;
            case 4:
                remove_person();
                getchar();
                getchar();
                break;
            case 5:
                update_person();
                getchar();
                getchar();
                break;
            case 6:
                remove_all();
                getchar();
                getchar();
                break;
            default :
                system("cls");
                printf("Thanks  for  using  phonebook  visit
again : )\n");
                getchar();
                getchar();
                exit(1);
        }
    }
}

// This will print main menu.
```

```
void print_menu()
{
    system("cls");

printf("\t\t**************************************************
***************\n");
    printf("\t\t*                    Welcome TO My phone book
*\n");

printf("\t\t**************************************************
***************\n\n");
    printf("\t\t\t1) list record\n\n");
    printf("\t\t\t2) Add person\n\n");
    printf("\t\t\t3) Search person Details\n\n");
    printf("\t\t\t4) Remove person\n\n");
    printf("\t\t\t5) Update person\n\n");
    printf("\t\t\t6) Delete all contacts\n\n");
    printf("\t\t\t7) exit Phonebook\n\n\n");

    printf("\t\t\t\tEnter your Choice : ");
}

// This function will add contact into phonebook.
void add_person()
{
     //no records
    system("cls");
    FILE *fp;
    fp = fopen("phonebook_db", "ab+");
    if (fp == NULL)
    {
        printf("Error in file opening, Plz try again !\n");
        printf("Press any key to continue....\n");
        return;
    }
```

```
    else
    {
        person p;
        take_input(&p);
        fwrite(&p, sizeof(p), 1, fp);
        fflush(stdin);
        fclose(fp);
        system("clear");
        printf("Record added Successfully\n");
        printf("Press any key to continue ....\n");

    }
}

// By this we take contact information.
void take_input(person *p)
{
    system("cls");
    // This getchar is for taking \n occured by previous
functions.
    // So that scanset in scanf will work properly.
    getchar();
    printf("Enter name : ");
    // Here we are using scanset '^' - >  until get
    scanf("%[^\n]s",p->name);

    printf("Enter country code : ");
    scanf("%s",p->country_code);

    printf("Enter mobile no : ");
    scanf("%s",p->mble_no);

    printf("Enter sex : ");
    scanf("%s",p->sex);

    printf("Enter email : ");
```

```
    scanf("%s",p->mail);
}

// This function will list contact available in phonebook.
void list_record()
{
     //no records
    system("cls");
    FILE *fp;
    fp = fopen("phonebook_db", "rb");
    if (fp == NULL)
    {
        printf("Error in file opening, Plz try again !\n");
        printf("Press any key to continue....\n");
        return;
    }
    else
    {
        person p;

printf("\n\t\t\t\t*****************************************
**********************************\n");
        printf("\t\t\t\t*                          Here is all
records listed in phonebook                    *\n");

printf("\t\t\t\t*****************************************
**********************************\n\n\n");
        printf("   NAME\t\t\t\t    COUNTRY CODE\t\t     PHONE
NO\t\t    SEX\t\t             EMAIL\n");
        printf("-----------------------------------------
-------------------------------------------------------
-------------------------------------\n");
        // fseek(fp,-(sizeof(p)*2L),2);
        while (fread(&p, sizeof(p), 1, fp) == 1)
        {
            int i;
```

```
            int len1 = 40 - strlen(p.name);
            int len2 = 19 - strlen(p.country_code);
            int len3 = 15;
            int len4 = 21 - strlen(p.sex);
            printf("%s",p.name);
            for(i=0;i<len1;i++) printf(" ");

            printf("%s",p.country_code);
            for(i=0;i<len2;i++) printf(" ");

            printf("%s",p.mble_no);
            for(i=0;i<len3;i++) printf(" ");

            printf("%s",p.sex);
            for(i=0;i<len4;i++) printf(" ");

            printf("%s",p.mail);
            printf("\n");
            fflush(stdin);
        }
        fflush(stdin);
        fclose(fp);
        printf("\n\nPress any key to continue....\n");

    }
}

// This function will search contact in phonebook.
void search_person()
{
    system("cls");
    char phone[20];
    printf("Enter Phone number of the person you want to
search : ");
    scanf("%s",&phone);
```

```
    FILE *fp;
    fp = fopen("phonebook_db", "rb");
    if (fp == NULL)
    {
        printf("Error in file opening, Plz try again !\n");
        printf("Press any key to continue....\n");
        return;
    }
    else
    {
        int flag = 0;
        person p;
        while (fread(&p, sizeof(p), 1, fp) == 1)
        {
            if(strcmp(p.mble_no,phone)==0)
            {
                printf("   NAME\t\t\t\t    COUNTRY  CODE\t\t
PHONE NO\t\t    SEX\t\t             EMAIL\n");
                printf("-----------------------------------
-------------------------------------------------------------
-----------------------------------------------\n");
                int i;
                int len1 = 40 - strlen(p.name);
                int len2 = 19 - strlen(p.country_code);
                int len3 = 15;
                int len4 = 21 - strlen(p.sex);
                printf("%s",p.name);
                for(i=0;i<len1;i++) printf(" ");

                printf("%s",p.country_code);
                for(i=0;i<len2;i++) printf(" ");

                printf("%s",p.mble_no);
                for(i=0;i<len3;i++) printf(" ");

                printf("%s",p.sex);
```

```
                for(i=0;i<len4;i++) printf(" ");

                printf("%s",p.mail);
                printf("\n");

                flag = 1;
                break;
            }
            else continue;
            // fflush(stdin);
        }
        if(flag == 0)
        {
            system("cls");
            printf("Person is not in the Phonebook\n");
        }
        fflush(stdin);
        fclose(fp);
        printf("\n\nPress any key to continue....\n");
    }

}

// This function will remove contact from phonebook.
void remove_person()
{
    system("cls");
    char phone[20];
    printf("Enter Phone number of the person you want to
remove from phonebook : ");
    scanf("%s",&phone);

    FILE *fp,*temp;
    fp = fopen("phonebook_db", "rb");
    temp = fopen("temp","wb+");
    if (fp == NULL)
```

```
    {
        printf("Error in file opening, Plz try again !\n");
        printf("Press any key to continue....\n");
        return;
    }
    else
    {
        person p;
        int flag = 0;
        while (fread(&p, sizeof(p), 1, fp) == 1)
        {
            if(strcmp(p.mble_no,phone)==0)
            {
                system("cls");
                printf("Person removed successfully\n");
                flag = 1;
            }
            else fwrite(&p,sizeof(p),1,temp);
            fflush(stdin);
        }
        if(flag == 0)
        {
            system("cls");
            printf("No record found for %d number\n",phone);
        }
        fclose(fp);
        fclose(temp);
        remove("phonebook_db");
        rename("temp","phonebook_db");
        fflush(stdin);
        printf("Press any key to continue....\n");

    }

}
```

```
// This function will update contact information.
void update_person()
{

    system("cls");
    char phone[20];
    printf("Enter Phone number of the person you want to
update details : ");
    scanf("%s",&phone);

    FILE *fp,*temp;
    fp = fopen("phonebook_db", "rb");
    temp = fopen("temp","wb+");
    if (fp == NULL)
    {
        printf("Error in file opening, Plz try again !\n");
        printf("Press any key to continue....\n");
        return;
    }
    else
    {
        int flag = 0;
        person p;
        while (fread(&p, sizeof(p), 1, fp) == 1)
        {
            if(strcmp(p.mble_no,phone)==0)
            {
                take_input(&p);
                fwrite(&p, sizeof(p), 1, temp);
                system("clear");
                printf("Data updated successfully\n");
                flag = 1;
            }
            else fwrite(&p,sizeof(p),1,temp);
            fflush(stdin);
        }
```

```
        if(flag == 0)
        {
            system("cls");
            printf("No record found for %d number\n",phone);
        }
        fclose(fp);
        fclose(temp);
        remove("phonebook_db");
        rename("temp","phonebook_db");
        fflush(stdin);
        printf("Press any key to continue....\n");
    }
}

// This function will clear all the data of phonebook.
void remove_all()
{
    char choice;
    system("cls");
    remove("./phonebook_db");
    printf("All    data    in    the    phonebook    deleted
successfully\n");
    printf("Press any key to continue ... \n");
}
```