



Civinnovate

Discover, Learn, and Innovate in Civil Engineering



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS**

MINI - PROJECT

BOOK -RECORD SYSTEM

SUBMITTED BY:

SUBMITTED TO:

ELECTRONICS & COMPUTER DEPARTMENT

**DATE:
2067/03/02**

TABLE OF CONTENTS

PAGE NO:

Acknowledgement-----1
Background-----2
General Theory-----3
Source Code-----8
Discussion-----16
Output or Result-----18
Summary-----21
Reference-----22

ACKNOWLEDGEMENT

We would like to deeply thank the various people who provided us with the useful and helpful assistance for the preparation of this project.

First, we would like to thank Mr. Surendra Shrestha, our teacher of computer programming-I who gave us the concepts of C programming language. Also we would like to thank all our dear friends and seniors for their advice, encouragement, support and help.

Lastly, we would like to thank authors of various books in C language which provided us great help for the preparation of this project.

- Mr. Yashwant Kanetkar-Let Us C.
- Mr. Venugopal-Programming With C.
- Mr. Byron S. Gotterfried-Programming With C.
- Mr. Balagurusamy-Programming In ANSI C.

Background

For technical students, the most important factor to measure their ability and skill is their practical performance rather than their theoretical knowledge. Considering this fact, an assignment is given to the students of electrical-1 to prepare a project using C language in order to develop their practical ability to develop programs and software using c programming language as per requirements.

This project is a part of the subject Computer Programming –I, first year and first part of Electrical Engineering course.

This project that we have worked on is based on the management of customer accounts in different banks.

To build the project, it took us nearly two weeks. The project required vast knowledge of use of structure and data file which are two important and complex features of C programming language. We got help from all the lecture classes that happened throughout the semester being held by our computer teacher, associate professor Mr. Surendra Shrestha. We also were guided by our computer lab teachers. Rest, we consulted various books on c programming that turned out to be more than useful in gaining concepts of computer programming.

After gaining the useful concepts, we started the project on modular level. Each of our team members shared equal load by writing different parts of the program such as different user-defined functions. We would devote about one hour time each day during the college days and worked throughout the day during holidays.

After the program was successfully written, compiled and tested for syntactic and logical errors in about two weeks, we had then to prepare the remaining parts of the project such as discussion and other theoretical part. We consulted our English teacher, Raj Kumar Yadav for helping us with the rules and procedure of writing a project. After getting valuable suggestions from him, we started writing the theoretical part. In this case also, we divided the work load among ourselves. This kind of working as a team and working hard marked the completion of the project.

Objectives:-

1. To learn about different library functions included in different header files.
2. To learn about the use of user defined function, structure, array and data file in C.
3. To learn to be able to develop complex programs aimed at solving particular task in practical field as per users requirements.
4. To be able to work in group as a team sharing different responsibilities.

General Theory

This project is based on high level language i.e. c programming. In this project we use important parts of c programming which are control statement, looping, function, array, structure, pointer, data file.

C programming language:

c is structured programming based computer programming language was developed by Dennis Ritchie at Bell laboratories in 1972. Structured programming refers to programming that produce program with clean flow, clear and a degree of modularity or hierarchical structure is a simple, contained, versatile, excellent, efficient, fast general purpose language. It has high degree of language of C is a function oriented additional task including input and output, graphics, math computation and access to peripheral devices are placed as library function.

Control Statement:

Logical operation is carried out by several symmetrical or logical statements. There are two types of control statement based on their function.

Selective structure:

Selective structures are used when we have a number of situations where we need to change the order of execution of statements based on certain condition. The selective statements make a decision to take the right path before changing the order of execution. C provides the following statements for selective structure:

if statements

switch statements

if statements:

The if statement is a powerful decision making statement and it is used to control the flow of execution of statements. It is a two way statement and is used in conjunction with an expression.

If statement allows the computer to evaluate the expression first and then on depending whether the value of the expression is true or false it transfer the control to the particular statement. At this point of the program has two paths to follow: one for true condition and other for false condition. The types of if statements are explained below:

Simple if statement:

The simple if statement is used to conditionally excite a block of code based on whether a test condition is true or false. If the condition is true the block of code is executed, otherwise it is skipped. The syntax of if statement is given below:

```
if(test expression)
{
    statement-block;
}
statement-x;
```

if else statement

The if else statement extends the idea of the if statement by specifying another section of code that should be executed only if the condition is false i.e. conditional branching. True- block statements are to be executed only if the test expression is true and false block statements to be executed only if the condition is false. The syntax of if else statement is given below:

```
if(test expression)
{
    true block statement;
}
else
{
    false block statement;
}
```

The switch statement:

c has built in multi way decision statement known as switch. It successively test the value of an expression against a list of case values (integer or character consonants).when a match is found the statement associated with that case is executed. The syntax of switch expression is given below:

```
switch(expression)
{
    case constant-1:
        block-1;
        break;
    case constant-2:
        block-2;
        break;
    .....
    .....
    case constant-2:
        block-n;
        break;
    default:
        default statement:
}
```

Looping:

Loop caused a section of code to be repeated for a specified number of times or until some condition holds true. When a condition becomes false, the loop terminates and control passes to statement below loop. Different types of loops are discussed below with their major characteristics and syntax used in C:

While loop:

The while loop specifies that a section of code should be executed while a certain condition holds true. The syntax of while loop is given below:

```
while(test expression)
{
    body of loop
(
statements block)
}
```

do while statement:

the do while statement is very similar to while statement. It also specifies that a section of code should be executed while a certain condition holds true. the difference between while and do while loop is that while loop test its condition at the top of its loop but do while loop tests its condition at the bottom of loop. In while loop, if the test condition is false, the block of code is skipped. Since condition is tested at the bottom of loop in do while loop, its block of code is always executed at least once. The syntax of do while loop is given below:

```
do
{
    body of loop
}while (test expression);
```

For loop:

the for loop is used to execute a block of code for a fixed number of repetitions. Initialization is generally an assignment statement used to set loop control variable. Test expression is a relational expression that determines when loop exits. Update expression defines how the loop variable changes each time when the loop is repeated. The syntax of for loop is given below:

```
for(initialization expression;test expression;update expression)
{
    body of loop;
}
```

break statement:

The break statement is used to jump out of loop. The break statement terminates the execution of the nearest enclosing loop. Control passes to the statement that follows the

terminated statement. in a switch break statement causes the program to execute the next statement after switch.

break;

Function:

A function is a self contained program segment that carries out some specific well defined task. Every c program consists of one or functions. Execution of program always begins by carrying out instruction in main. Function makes program significantly easier to understand and maintain. A well written function may be reused in multiple programs. Program that are easier to design, debug and maintain.

Return statement:

A function may or may not send back any value to the calling function. If it does, it is through return statement. The called function can only return only one value per call at most. The syntax of return statement is given below:

Return;

Pointer:

A pointer is a variable that represents the location (rather than value) of a data item, such as a variable or an array element. Pointers can be used to pass information back and forth between a function and a reference point. Pointer provides a way to return multiple data items from a function via function argument. When a pointer variable is declared, the variable name must be preceded by an asteric (*).the syntax of a pointer declaration is:

data type *ptar;

Structure:

It is a heterogeneous user defined data type. It is also called constructed data type. It may contain different data types .Structure can also store non homogenous data type into a single collection. Structure may contain pointet, arrays, or even other structures other than the common data types such as int, float, long int etc. A structure provides a means of grouping variables under a single name for easier handling and identification. It can be defined as new named types. It is a convenient way of grouping several pieces of related information together. Complex hierarchies can be created by nesting structures. Structures may be copied to and assigned. They are also useful in passing groups of logically related data into structures. The declaration of structures is given below:

struct tag

```
{  
    member 1;  
    member 2;  
  
    member n;  
};
```

File:

Many applications require that information be written to or read from an auxiliary memory device. Such information is stored on the memory device in the form of a data file. The data files allow us to store information permanently and to access and alter that information whenever necessary.

Opening a file:

Before performing any input / output operation, file must be opened. While opening file, the following must be specified:-

- i) The name of file.
- ii) The manner in which it should be opened (that for reading ,writing ,both reading and writing ,appending at the end of file, overwriting the file)
- iii) when working with a stream oriented data file ,the first step is to establish a buffer area, where information is temporary stored while being transferred between the computers memory and data file .the buffer area is established by writing

FILE *ptvar;

where File is a special structure type establishes the buffer area and ptvar is a pointer variable that indicates the beginning of the buffer area the library function fopen is used to open a file .This function is used to open a file .This function is typically written as

ptvar=fopen(file name, file type);

where file name and file type are strings that represent the name of the data file and the manner in which the data file will be utilized.

Finally, a file can be closed at the end of the program. This can be accomplished with the library function fclose. The syntax is simply,

fclose(ptvar);

Processing a file:

Most data file application requires that a data file be altered as it is being processed. For example, in an application involving the processing of customer records, it may be desirable to add new records to the file. To delete the existing records, to modify the contents or to rearrange the records.

File types:

DOS treats files in two different ways that as binary or text file. Files almost all UNIX systems do not make any distinction between the two. If a file is specified as the binary type, the file I /O functions do not interpret the contents of the file when they read from, or write to the file. But if the file is specified as the text type, the file I / O functions interpret the contents of the file. The basic differences in these two types of files are:

- i) ASCII 0*1a is considered as end of file character by the file I /O functions when reading from a text file and they assume that the end of the file has been reached.
- ii) In case of DOS, a new line is stored as the sequence 0*0a on the disk in case text files.

SOURCE CODE

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
void input();
void writefile();
void search();
void output();
void modify();
void del();

struct date{
    int month;
    int day;
    int year;
};

struct account {
    int number;
    char name[100];
    int acct_no;
    float mobile_no;
    char street[100];
    char city[100];
    char acct_type;
    float oldbalance;
    float newbalance;
    float payment;
    struct date lastpayment;
}customer;
long int tl,sl,ts;

void main()
{
    int i,n;
    char ch;
    clrscr();
```

```
_setcursortype(_NOCURSOR);
printf(" CUSTOMER ACCOUNT MANAGEMENT SYSTEM:\n\n");
printf("=====\n");
printf("*****\n");
printf("\n1:      to add account on list\n");
printf("2:      to search customer account\n");
printf("3:      to modify existing account\n");
printf("4:      to delete existing account\n");
printf("5:      exit\n");
printf("\n*****\n");
printf("=====\n");
do{
    fflush(stdin);
    printf("\nselect what do you want to do?");
    ch=getche();
}while(ch<='0' || ch>'5');
switch(ch){
    case '1':
        clrscr();
        printf("\nhow many customer accounts?");
        scanf("%d",&n);
        for(i=0;i<n;i++){
            input();
            if(customer.payment>0)
                customer.acct_type=(customer.payment<0.1*customer.oldbalance)? 'O': 'D';
            else
                customer.acct_type=(customer.oldbalance>0)?'D' : 'C';
            customer.newbalance=customer.oldbalance -
customer.payment;
            writefile();
        }
        main();
    case '2':
        clrscr();
        printf("search by what?\n");
        printf("\n1 --- search by customer number\n");
        printf("2 --- search by customer name\n");
        search();
        ch=getche();
        main();
    case '3':
        modify();
        main();
```

```
        case '4':
            del();
            main();
        case '5':
            clrscr();
            delay(700);
            gotoxy(25,25);
            cprintf("\nA PROJECT BY TAPAN,SUNIL,BIWAS &
RAVINDRA");
            delay(1500);
            exit(1);
    }
}
```

```
void input()
{
    FILE *fp=fopen("bidur.dat","rb");
    fseek (fp,0,SEEK_END);
    tl=ftell(fp);
    sl=sizeof(customer);
    ts=tl/sl;
    fseek(fp,(ts-1)*sl,SEEK_SET);
    fread(&customer,sizeof(customer),1,fp);
    printf("\ncustomer no:%d\n",++customer.number);
    fclose(fp);
    printf("    Account number:");
    scanf("%d",&customer.acct_no);
    printf("\n    Name:");
    fflush(stdin);
    scanf("%s",customer.name);
    printf("\n    mobile no:");
    scanf("%f",&customer.mobile_no);
    printf("    Street:");
    fflush(stdin);
    scanf("%s",customer.street);
    printf("    City:");
    fflush(stdin);
    scanf("%s",customer.city);
    printf("    Previous balance:");
    scanf("%f",&customer.oldbalance);
    printf("    Current payment:");
    scanf("%f",&customer.payment);
}
```

```
printf("    Payment date(mm/dd/yyyy):");
scanf("%d/%d/%d",&customer.lastpayment.month,&customer.lastpayment.day,&customer.lastpayment.year);
    return;
}
```

```
void writefile()
{
    FILE *fp;
    fp=fopen("bidur.dat","ab");
    fwrite(&customer,sizeof(customer),1,fp);
    fclose(fp);
    return;
}
```

```
void search()
{
    char ch;
    char nam[100];
    int n,i,m=1;
    FILE *fp;
    fp=fopen("bidur.dat","rb");
    do {
        printf("\nenter your choice:");
        ch=getche();
    } while(ch!='1' && ch!='2');
    switch(ch){
        case '1':
            fseek(fp,0,SEEK_END);
            tl=ftell(fp);
            sl=sizeof(customer);
            ts=tl/sl;
            do {
                printf("\nchoose customer number:");
                scanf("%d",&n);
                if(n<=0 || n>ts)
                    printf("\nyour input is incorrect!\n");
                else {
                    fseek(fp,(n-1)*sl,SEEK_SET);
                    fread(&customer,sl,1,fp);
                    output();
                }
            }
            printf("\n\nsearch again?(y/n)");
            ch=getche();
        }
}
```

```
    }while(ch=='y');
    fclose(fp);
    break;
case '2':
    fseek(fp,0,SEEK_END);
    tl=ftell(fp);
    sl=sizeof(customer);
    ts=tl/sl;
    fseek(fp,(ts-1)*sl,SEEK_SET);
    fread(&customer,sizeof(customer),1,fp);
    n=customer.number;

    do{
        printf("\nenter the name:");
        fflush(stdin);
        scanf("%s",nam);
        fseek(fp,0,SEEK_SET);
        for(i=1;i<=n;i++)
        {
            fread(&customer,sizeof(customer),1,fp);
            if(strcmp(customer.name,nam)==0)
            {
                output();
                m=0;
                break;
            }
        }
        if(m==1)
            printf("\n\nThis account doesn't exist\n");
            printf("\nsearch another?(y/n)");
            ch=getche();
    }while(ch=='y');
    fclose(fp);
}
return;
}
```

```
void output()
{
    printf("\n\n Customer no  :%d\n",customer.number);
    printf(" Name      :%s\n",customer.name);
    printf(" Mobile no   :%.f\n",customer.mobile_no);
}
```

```
printf(" Account number :%d\n",customer.acct_no);
printf(" Street      :%s\n",customer.street);
printf(" City        :%s\n",customer.city);
printf(" Old balance  :%.2f\n",customer.oldbalance);
printf(" Current payment:%.2f\n",customer.payment);
printf(" New balance   :%.2f\n",customer.newbalance);
printf(" Payment date
:%d/%d/%d\n\n",customer.lastpayment.month,customer.lastpayment.day,customer.lastpa
yment.year);
printf(" Account status :");
```

```
switch(customer.acct_type)
{
case 'C':
    cprintf("CURRENT\n\n");
    break;
case 'O':
    cprintf("OVERDUE\n\n");
    break;
case 'D':
    cprintf("DELINQUENT\n\n");
    break;
default:
    cprintf("ERROR!\n\n");
}

return;
}
void modify()
{
FILE *fp;
char name[100];
int i=1;
long int size=sizeof(customer);
if((fp=fopen("bidur.dat","rb+"))==NULL)
exit(1);
printf("\nEnter name of account to modify:");
fflush(stdin);
scanf("%s",name);
while(fread(&customer,sizeof(customer),1,fp)==1)
{
if(strcmp(customer.name,name)==0)
```



```
{
    i=0;
    printf("\n Enter new data:\n");
    printf("    Account number:");
    scanf("%d",&customer.acct_no);
    printf("\n    Name:");
    fflush(stdin);
    scanf("%s",customer.name);
    printf("\n    mobile no:");
    scanf("%f",&customer.mobile_no);
    printf("    Street:");
    fflush(stdin);
    scanf("%s",customer.street);
    printf("    City:");
    fflush(stdin);
    scanf("%s",customer.city);
    printf("    Previous balance:");
    scanf("%f",&customer.oldbalance);
    printf("    Current payment:");
    scanf("%f",&customer.payment);
    printf("    Payment date(mm/dd/yyyy):");

    scanf("%d/%d/%d",&customer.lastpayment.month,&customer.lastpayment.day,&
customer.lastpayment.year);
    if(customer.payment>0)

        customer.acct_type=(customer.payment<0.1*customer.oldbalance)? 'O': 'D';
        else
            customer.acct_type=(customer.oldbalance>0)?'D' : 'C';
            customer.newbalance=customer.oldbalance -
customer.payment;
            fseek(fp,-size,SEEK_CUR);
            fwrite(&customer,sizeof(customer),1,fp);

            break;
    }
}
if(i==1)
printf("Account not present");
else if(i==0)
printf("Account successfully modified");
getch();
```

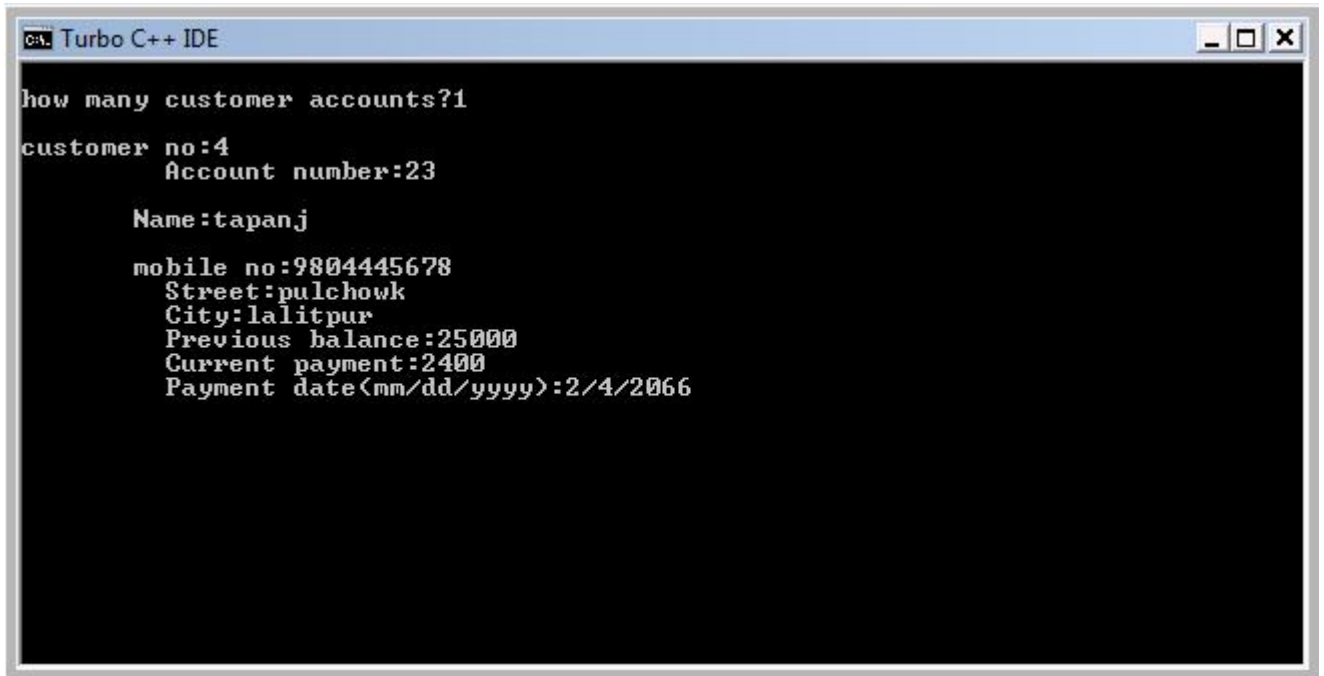
```
    fclose(fp);
}
void del()
{
    FILE *fp,*temp;
    char name[100];
    int i=1;
    if((fp=fopen("bidur.dat","rb"))==NULL)
        exit(1);
    if((temp=fopen("tempfile.dat","wb"))==NULL)
        exit(1);
    printf("\n    Enter name of account to delete:");
        fflush(stdin);
    scanf("%s",name);

    while(fread(&customer,sizeof(customer),1,fp)==1)
    {
        if(strcmp(customer.name,name)==0)
        {
            i=0;
            continue;
        }
        else
            fwrite(&customer,sizeof(customer),1,temp);
    }
    remove("bidur.dat");
    rename("tempfile.dat","bidur.dat");
    fclose(fp);
    fclose(temp);
    if(i==0)
        printf("Account successfully deleted");
    else if(i==1)
        printf("Account not present");
    getch();
}
```



```
CA Turbo C++ IDE
CUSTOMER ACCOUNT MANAGEMENT SYSTEM:
=====
*****
1:      to add account on list
2:      to search customer account
3:      to modify existing account
4:      to delete existing account
5:      exit
*****
=====
select what do you want to do?
```

Enter 1 to add new account



```
CA Turbo C++ IDE
how many customer accounts?1
customer no:4
    Account number:23
    Name:tapanj
    mobile no:9804445678
    Street:pulchowk
    City:lalitpur
    Previous balance:25000
    Current payment:2400
    Payment date(mm/dd/yyyy):2/4/2066
```

press enter to add to file and return to main screen

```
CA Turbo C++ IDE
search by what?
1 --- search by customer number
2 --- search by customer name

enter your choice:1
choose customer number:4

Customer no      :4
Name             :tapanj
Mobile no        :9804445696
Account number   :23
Street           :pulchowk
City             :lalitpur
Old balance      :25000.00
Current payment  :2400.00
New balance      :22600.00
Payment date     :2/4/2066

Account status   :OVERDUE

search again?(y/n)
```

press y to continue and n to return to main screen


```
CA Turbo C++ IDE
*****
1:      to add account on list
2:      to search customer account
3:      to modify existing account
4:      to delete existing account
5:      exit
*****
=====
select what do you want to do?3
Enter name of account to modify:tapanj

Enter new data:
  Account number:23

  Name:suraj

  mobile no:9804443456
  Street:lagankhel
  City:lalitpur
  Previous balance:34000
  Current payment:4500
  Payment date(mm/dd/yyyy):3/5/2066
```

press enter to complete modification and return to main screen



```
CA Turbo C++ IDE
CUSTOMER ACCOUNT MANAGEMENT SYSTEM:
=====
*****
1:      to add account on list
2:      to search customer account
3:      to modify existing account
4:      to delete existing account
5:      exit
*****
=====
select what do you want to do?4
      Enter name of account to delete:suraj
```

press enter to complete deletion

DISCUSSION

This mini-project is about billing system in a bank. We used control statement, looping, array, pointer, structure files etc. this program contains three header files such as stdio.h, conio.h and stdlib.h. This program stores customer's information, bank account, deposited

Account and payment. This program first ask to users asking what do you want to do with three options such as to add account, to search account, deleting the existing account and exit the program. The program begins by defining the composition of each customer's record and stream pointer fpt. We define the structure account using different variables such as name, number, mobile number, city, street, new balance; old balance etc. payment date is defined using another structure variable of account type. Similarly t1, s1 and ts are defined as global integer variable.

In main function, we define I, n as integer variable and ch as character variable. Here we prepare the first user interface of program which ask user to enter which for adding new record, searching, modify or deleting existing record.

ADDING NEW ACCOUNT

In our account management system we need to add new accounts for new customers. According to the input given for the no of new accounts to be created, we enter in a loop where we call input() function. In function input(), we open a data file tapan.dat in read mode. After that, with the help of library function fseek, we read the last block of record in the file and determine the customer number for a new customer number by incrementing the last customer number by 1. After this file is closed and user is promoted to input information of new account such as account number, name, address, new balance etc. then the control is returned to main function where customer account type is manipulated and another function write file is called. This function again opens the file in append mode and the customer records are written into the file using library function fwrite.

SEARCHING FOR AN ACCOUNT

We can search for existing account in two ways- first by reading customer number and second by read customer name. Here search() function is used to search the data in the user defined function search(), tapan.dat file is opened in read mode based on choice by user; search operation is carried out using number or name. For the first case, customer number is read from the user. If this number is greater than the number of existing account or less than 0, error message is printed. Otherwise, fseek function is used to locate the file pointer at the beginning of the required record and data is read using fread function. Now user define function output is called where all the available record of the account is printed on screen.

MODIFYING EXISTING RECORD

In user defined function modify(), tapan.dat file is opened in read mode. Here name of data file is written which is to be modified. fflush is used in this function to remove of the account, new data is written just same as the previous way. fseek function is used to locate the file pointer at the current position just at the back of the size of the customer and fwrite function is used to modify.

DELETING THE UNWANTED RECORD

In the function del(), in tapan.dat file is opened in read mode. In this function another file named tempfile.dat is also opened in write mode. Firstly the name of the account to be deleted is written. The customer name is compared with the typed name. All other data except that file is copied to the new file tempfile.dat and at last the tempfile.dat is renamed to tapan.dat and at last the file is closed. This is the most efficient method of deleting records from the file.

SUMMARY

This mini-project is prepared by c-programming using simple logic available in high level language of this programming. This program is concerned in billing system which can be used in banking issues. We have included different logic and statement such as switch statement, and if, if else as control statement and looping like: for, while and do while statement. The whole program is composition of array which support the use of a group of similar data type, pointer which is a variable that represent the location (rather than value) of data item such as a variable or an array element. This is very useful in c for low consumption of memory. Also we used strings which are a group of similar array data types. Similarly we use structure and files which provides a means of grouping variables under a single name for easier handling and identification. It can be also called constructed data type. It is a heterogeneous data type which may contain different data type in one time. Files allow us to store information permanently which can be modifying whenever necessary. Therefore a file is a place where a group of related data is stored.

In the beginning of program we can see different option such as adding account, searching account, deleting existing account and exit the program for which we have to enter the number shown in the option. We can add the new customer account which requires different information of customer such as name, mobile number, city, street, account number, payment date, old balance, new balance, account status etc. we can search the customer information about account status such as current balance, current payment by two way such as by name of customer and by number of account. If no match was found than again(y/n) will print where if we enter y than we can search for account information else we enter n than it returns to initial part of the program.

This program will be very beneficial in the banks for customer billing system in a systematic way. One can easily maintain the customer accounts, add huge number of accounts in the list, can easily search their accounts either by name or account number and can also change the accounts or delete the unwanted accounts of the banks.

Through this program we can also find out what sorts of account the customers hold i.e. whether customer account is overdue, current or delinquent. If the customer payment is less than ten percent of the customer old balance then the type of the customer account type is overdue otherwise it is deliquescent. Similarly if the customer old balance is greater than zero rupees than the account type is current. Customer new balance is calculated by subtracting customer old balance by customer payment. In this way the account status of the customer can be calculated. This program also provides the information about the customer payment date. These all information about the customers in the banks can be maintained easily and saved which provide the detail information about the customer.

REFERENCE

Learning c by examples

Krishna Kandel

Programming with C

Byron s. Gottfried

Let us C

Yashavant p.Kanetkar

Programming in ANSI C

E Balagurusamy

A book on C

Al Kelley / Ira pohl

Communication English

R K Yadav