**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, this mini-project is included in our lab assignment to prepare a project using C language in order to develop their practical ability to develop programs and software using c programming language .

In this project our project topic BOOK-RECORD SYSTEM .In this programm stores the data related about book record in any library.So this program can be used as library record system.

To build the project, it took us nearly one week. 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 senior student and teachers in lab. 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 2 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 one week, we had then to prepare the remaining parts of the project such as discussion and other theoretical part.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 aesteric (\*).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:-

1. The name of file.
2. 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)
3. 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\_type meaning**

**“r”** open an existing file for reading only.

**“w”** open a new file for writing only. If the file with specified file\_name currently exists it will be destroyed and new file is created in its place.

**“a”** open an existing file for appending. A new file will be created if the file with the specific file\_name does not exist.

**“r+”** open an existing file for both reading and writing.

**“w+”** open a new file for both reading and writing. If a file with the specified file\_name currently exists, it will be destroyed and a new file is created in its place.

**“a+”** open an existing file for reading and writing. A new file will be created if the file with the specified file\_name does not exist.

**THE FREAD AND FWRITE FUNCTIONS :**

Some applications involve the use of data files to store blocks of data, where each block consists of a fixed number of contiguous bytes. Each block will generally represent a complete data structure, such as a structure or an array. for such applications it may be desirable to read the entire block from the data file or write the entire block to the file.

The library function fread and fwrite are intended to be used in situations of this size of the data block, the number of data block being transferred and file pointer. thus typical fwrite and fread functions :

**Fwrite(&customer, sizeof(record),1,fpt);**

**Fread(&customer, sizeof(record),1,fpt)**

Where customer is a structure variable of type record and fpt is the file pointer associated with the data file that has been opened for input/output. Once an unformatted data file has been created with fwrite, it can be read with fread function. The function returns a zero value if an end-of-file condition has been detected and non-zero value if an end-of-file is not detected. Hence, a program that reads an unformatted data file can be reading file, as long as the value returned by fread is non-zero value.

**rewind :**

rewind takes a file pointer and resets the position to the start of the file. For example the statement :

**rewind(fp);**

**N=ftell(fp);**

Would assign to fp because the file position has been set to the start of the file by rewind. This function helps us in reading a file more than once. Without having to close and open the file

**Algorithm**

1.Start

2.Declare structure library1 containing character book\_title, author\_name ,integer accesssion number,character caretegory and structure test containing char name1

3.Declare file pointer \*f,\*fp

4.Declare function prototypes of

[1] add book record

[2]display book record

[3 delete book record

[4] Search book record

[5]Modify book record [6]Exit

[7]link

1. Declare character filename and integer count
2. **In the main function**

=>Declare variables integer choice and character check

=>Enter the main menu

=>Enter your choice and read choice

=>Use Switch in which case values are used as argument and following casses may occur according the choice

1. Call add book record function
2. Call display record function
3. Call delete book record function
4. Call search book record function
5. Call modify book record function

6. Call exit function

=> If all above case values doesnot match set a default statement and if default is encountered then goto main menu

**7.Add book record function definition()**

=>declare variables test as char type .

=>open the file in append and write mode

=>if file cannot open then exit

else enter the book records

=>get the book title ,author name,acc no& category of book

=>write to the file.

=>if want to exit then press esc key

=>close that file.

**8.display book record function definition()**

=>open the file in read mode.

if the file doesnot exist print file canot be open the file

=> read all the file record

=> display all book records and goto main menu

=>close that file.

**9.delete book record function definition()**

=>enter the name of book to be deleted

=>open the file in rb+ mode

=>if file cann’t open ,print cannot open the file and exit

=>open new tamporary file in wb mode

=>if canot open the file then exit

=>read file

=>check for name of structure library book title and enetered name

-If true continue

- else write the elements of library to new file

=> remove old file

=> rename newfile with oldfile

=>close the both files

**10.search book record function definition()**

**=>**open the file in rb+ mode

-if cannot open then exit

=>here we can search the book record information on the basis of book name ,author name,accession number of book& category of book

=>use switch statement for choosing any of srarching type

=>for the paricular searching mode enter the the type ofelement of the structure

Library

=>read the library structure file

=>printf the record of book

=> close the file

**11.modify book record function definition()**

=> declare character name[100] and character another’Y’

=>open the file in rb+ mode

- if file canot open then exit

=>use while loop to take the name of book for modifying book record

Until another==’Y’

=>enter the name of book to modify the record of book

=>read the file using fread

=>compare the book\_title and name of book entered

If true enter the records of book

=>use fseek function to modify the record

=>write to the file new records

=>enter Y if want to modify another book record

=>close the file

**sourcecode**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<conio.h>

FILE \*f,\*fp;

int i;

void add\_book();

void disp\_book();

void delete\_book();

void search();

void modify();

float link();

void exit();

struct library1

{

char book\_title[100];

char author\_name[100];

int acc ;

char category[20];

}library;

struct test{

char name1[30];

}t;

int count;

char filename[30];

int main()

{

int choice;

char check;

strcpy(filename,"rakesh");

up:

clrscr();

while(1)

{ printf("\n\n\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\n\n\t\tMainmenu!!!");

printf("\n\n\n\t\t1.Add Book Information");

printf("\n\n\t\t2.Display Book Information");

printf("\n\n\t\t3.Delete Book Records");

printf("\n\n\t\t4.Search Specified Book");

printf("\n\n\t\t6.Modify Records");

printf("\n\n\t\t7.Exit");

printf("\n\n\n\n\t\t\t Your Choice:");

scanf("%d",&choice);

clrscr();

switch(choice)

{

case 1:

{

add\_book();

break;

}

case 2:

{

disp\_book();

getch();

break;

}

case 3:

delete\_book();

printf("\n\n\npress any key");

break;

case 4:

search();

printf("\n\n\n\n press any k");

break;

case 6:

modify();

printf("\n\n\n\npress any key");

break;

case 7:

exit(0);

default:

printf("ENter 1 2 3 4 5 6 7");

}

goto up;

}

}

void add\_book()

{ char test;

if((f=fopen(filename,"ab+"))==NULL)

{if((f=fopen(filename,"wb+"))==NULL)

{

printf(" canot open the file:");

exit(0);

getch();

}

}

while(1)

{

clrscr();

printf("\n\n\n\n\n\t\t\tEnter thr details of the book:\n");

fflush(stdin);

printf("\n\n\n\t\t\tName:");

scanf(" %[^\n]",library.book\_title);

printf("\n\t\t\tAuthor:");

scanf(" %[^\n]",library.author\_name);

printf("\n\t\t\tAccession no:");

scanf(" %d",&library.acc );

printf("\n\t\t\tCategory:");

scanf(" %s",library.category);

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

fflush(stdin);

printf("\n\n\n\t\t\t\tEnter esc key to exit");

test=getche();

if(test==27)

break;

}

fclose(f);

}

void disp\_book()

{

if((f=fopen(filename,"rb"))==NULL)

{

clrscr();

printf("Cannot open the file");

getch();

exit(0);

}

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

{

clrscr();

gotoxy(25,10);

printf("DETAILS OF BOOKS IN LIBRARY");

for(i=1;i<28;i++)

{

textcolor(i);

gotoxy(24+i,11);

cprintf("\*");

}

printf("\n\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s",library.book\_title,library.author\_name,library.acc ,library.category);

getch();

}

clrscr();

gotoxy(25,10);

printf("DETAILS OF BOOKS IN LIBRARY");

for(i=1;i<28;i++)

{

textcolor(i);

gotoxy(24+i,11);

cprintf("\*");

}

printf("\n\n\n\n\n\n\t\t\t::No Records Available::");

fclose(f);

}

void delete\_book()

{

FILE \*f,\*temp;

int flag;

char name[100];

gotoxy(20,10);

printf("Enter the name of the book to be deleted:\n\n\t\t\t\t");

fflush(stdin);

scanf(" %[^\n]",name);

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

if(f==NULL)

{

printf("\n can not open the file");

exit(1);

}

temp=fopen("tempfile","wb");

if(temp==NULL)

exit(1);

flag=1;

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

{

if((strcmp(library.book\_title,name))==0)

{

flag=0;

continue;

}

else

fwrite(&library,sizeof(library),1,temp);

}

remove(filename);

rename("tempfile",filename);

fclose(f);

fclose(temp);

gotoxy(25,20);

if(flag==0)

printf(" Record Deleted");

else

printf("::No such Record::");

getch();

}

void search()

{

FILE \*f;

char c,name[100],nam[100],z;

int flag,a,i=0;//j=0;

if((f=fopen(filename,"rb+"))==0)

exit(0);

printf("\n\n\n\n\t\t\tWhat do you know about the book?\n\n\n\n\t\t\t1. The name of the book(N)\n\n");

printf("\t\t\t2. The author of the book(A)\n\n\t\t\t3. The accesion code of the book(C)\n");

printf("\n\t\t\t4. The category of the book(T)\n\n\n\n\n\t\t\t\t\t");

scanf(" %c",&c);

clrscr();

switch(c)

{

case 'N':

case 'n':

printf("\n\n\n\n\t\tEnter the name of the book:");

fflush(stdin);

scanf("%[^\n]",name);

flag=1;

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

{

if((strcmp(library.book\_title,name))==0)

flag=0 ;

else

continue;

if(flag==0)

{

printf("\n\n\n\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s",library.book\_title,library.author\_name,library.acc ,library.category);

getch();

clrscr();

i++;

}

}

if(flag==1||i==0)

{

gotoxy(25,20);

printf("::No such Record::");

getch();

}

getch();

break;

case 'A':

case 'a':

printf("\n\n\n\n\t\tAuthor Name:");

fflush(stdin);

scanf("%[^\n]",name);

flag=1;

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

{

if((strcmp(library.author\_name,name))==0)

flag=0 ;

else

continue;

if(flag==0)

{

i++;

printf("\n\n\n\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s",library.book\_title,library.author\_name,library.acc ,library.category);

getch();

clrscr();

}

}

if(flag==1||i==0)

{

gotoxy(25,20);

printf("::No such Record::");

getch();

}

getch();

break;

case 'C':

case 'c':

printf("\n\n\n\n\t\tAccesion code:");

fflush(stdin);

scanf("%d",&a);

flag=1;

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

{

if(library.acc==a)

flag=0 ;

else

continue;

if(flag==0)

{

i++;

printf("\n\n\n\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s",library.book\_title,library.author\_name,library.acc ,library.category);

getch();

clrscr();

}

}

if(flag==1||i==0)

{

gotoxy(25,20);

printf("::No such Record::");

getch();

}

break;

case 'T':

case 't':

printf("\n\n\n\t\tWe have arranged the books in following categories:\n");

fp=fopen("name","wb");

fclose(fp);

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

{

fp=fopen("name","rb");

flag=1; fflush(stdin);

while(fread(&t,sizeof(t),1,fp))

{ fflush(stdin);

if(strcmp(library.category,t.name1)==0)

{

flag=0;

break;

}

}

fclose(fp);

fp=fopen("name","ab");

if(flag==1)

{

strcpy(t.name1,library.category);

fwrite(&t,sizeof(t),1,fp);

fflush(stdin);

}

fclose(fp);

}

fp=fopen("name","rb");

printf("\n");

while(fread(&t,sizeof(t),1,fp))

{

printf("\n\n\t\t\t\t=> %s",t.name1);

fflush(stdin);

}

fclose(fp);

printf("\n\n\t\t\tEnter the category:");

scanf("%[^\n]",name);

clrscr();

flag=1;

rewind(f);

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

{

if((strcmp(library.category,name))==0)

flag=0 ;

else

continue;

if(flag==0)

{

i++;

printf("\n\n\n\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s",library.book\_title,library.author\_name,library.acc ,library.category);

getch();

clrscr();

}

}

if(flag==1||i==0)

{

gotoxy(25,20);

printf("::No such Record::");

getch();

}

fclose(f);

break;

default:

gotoxy(30,10);

printf("WRONG ENTRY");

gotoxy(24,11);

printf("TRY ANY VALID CHARACTER\n");

break;

}

fclose(f);

}

void modify()

{

char name[100],a[100],b[100],d[50],another='Y';

int c;

if((f=fopen(filename,"rb+"))==NULL)

{

exit(0);

}

while(another=='Y')

{

printf("\n\n\n\n\t\t\tEnter the name of book to b modified:\n\n\t\t\t\t");

fflush(stdin);

scanf("%[^\n]",name);

rewind(f);

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

{ i=1;

if((strcmp(library.book\_title,name))==0)

{

printf("\n\n\t\t\t\t::New records::");

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

printf("\n\n\n\n\t\t\tBook Name: ");

fflush(stdin);

scanf("%[^\n]",library.book\_title);

printf("\n\t\t\tAuthor Name:");

fflush(stdin);

scanf("%[^\n]",library.author\_name);

printf("\n\t\t\tAccession No:",library.acc);

fflush(stdin);

scanf("%d",&library.acc);

printf("\n\t\t\tCategory:",library.category);

fflush(stdin);

scanf("%[^\n]",library.category);

fseek(f,(i-1)\*sizeof(library),SEEK\_SET);

fflush(stdin);

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

break;

}

i++;

}

printf("\n\n\n\n\n\n\t\t\tModify another records (Y/N)");

fflush(stdin);

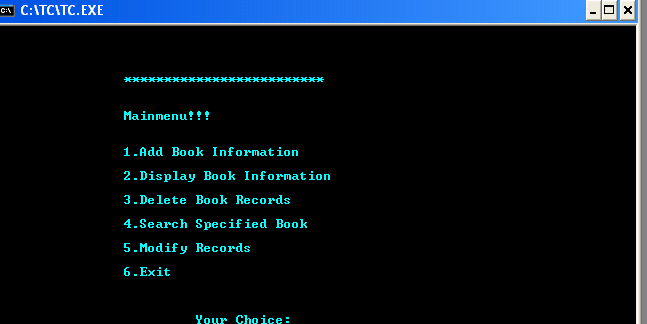
another=getche() ;

}

fclose(f);

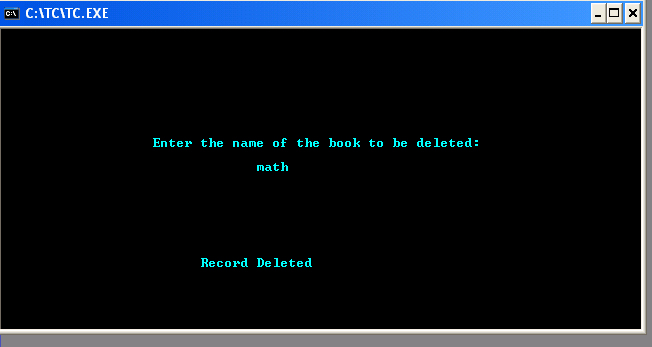
}

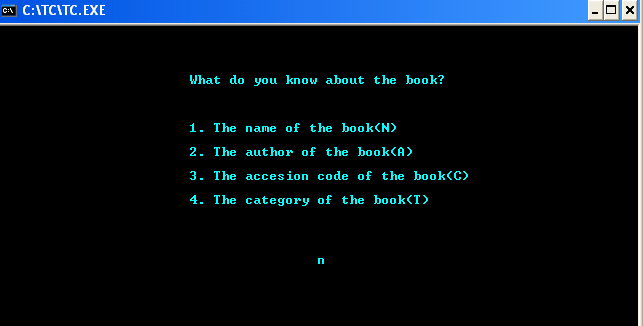
**output**

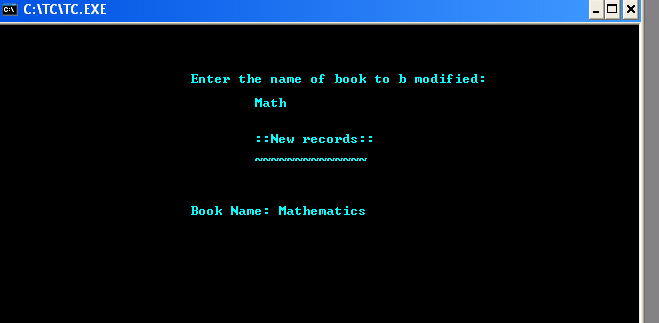












**DISCUSSION**

This mini-project is about **book record system**. We used control statement, looping, array, pointer, structure, files etc. this program contains many header files such as “stdio.h”, “conio.h”, “string.h”, “stdlib.h” . This program add, didplay, detelete ,search and modify the book record. This program first ask to users asking what do you want to do with 6 options such as to add book information ,dislaplay,delete,search,modify book information and exit the program.

During program compilation ,we faced different problems to handle individual tasks.In this program there was no error but there is some logical errors in delete,modify,search operations .For debug these logical errors we take help of teachers and senior student in hostel.By facing these difficulties we have knowledge of different new facts such as how and where proper use different mode in file handaling.During compilation we also solved the problem by group discussion.

In this way we successfully completed our project.

**SUMMARY**

Our main aim in this program is to prepare book record in library .The class modules and problems through done on the computers itself were much theoretical than practical because they were unable to teach us there application in our daily life.

The mini project on the other hand not only taught us practical use of program but also helped us in gaining skills like co-ordination, leadership, management of time, planning, foresights, etc. Above all it taught us that we could do something useful with c-programming. It gave taste of being a programmer which will very useful in our future.

Thus we can predict that we succeeded on our aim.The problems we faced make us more confident in C language.

**REFRENCECES**

ANSI C => E Balaguruswami

Prgramming in C => Baryon S Gotterfried

Lerning C by example => Krisna Kadel

Let us C => Yasvant Kanetkar

Mastering C => Venugopal

Guider => Senior students