**INTRODUCTION**

Welcome,

In this project we feel highly privileged to introduce you to a software design meant for library management

This project is useful for the libraries to issue books and keep records of the members and their details.

It can also be used to modify the records of members and books and also to add more books and members in the library.

**Flowchart**

MENU **mmmmMENU**

.

**Applications**

This program is 90% applicable in commercial field.

This program is totally efficient in terms of storing user detail, book detail, adding new records and appending them.

The project is efficient, user friendly and easy to handle.

We have tried our level best to make it complete software so that it can be used commercially.

**Header files used**

|  |  |  |
| --- | --- | --- |
| **Name of the Header file** | **Functions Used** | **Use Of The Functions** |
| **fstream.h** | 1.open()  2.close()  3.read()  4.write() | Opens the file for fstream  Closes the open file  Reads bytes from a file into buffer  Writes a buffer of data to a file or device |
| **conio.h** | 1.clrscr()  2.getch() | Clears the current text window  Reads a single character (without echoing) |
| **stdio.h** | 1.gets()  2.remove()  3.rename()  4.sizeof() | Collects the strings of character  Removes the data file  Renames the data file  Determine the length  of the strings |
| **string.h** | 1.strcpy()  2.strcmpi() | Copies string from 1 variable to another variable  Compares 2 strings without case sensitivity |
| **ctype.h** | 1.tolower() | Translates characters to lowercase |
| **iomanip.h** | 1.setw() |  |
| **dos.h** | 1.delay() | Used to delay |

**Classes and their functions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLASSES** | **FUNCTION OF CLASS** | | **MEMBER FUNCTIONS** | **FUNCTION OF MEMBER FUNCTIONS** |
| **MENU** | THIS CLASS CONTROL ALL THE FUNCTIONS IN THE MENU | | **public :**  void main\_menu(void) ;  void introduction(void) ;  **private :**  void edit\_menu(void) ;  void edit\_book(void) ;  void edit\_member(void) ; | FUNCTION TO DISPLAY MAIN MENU AND CONTROL ALL THE FUNCTION  IN THE MAIN MENU.  FUNCTION TO DISPLAY THE INTRODUCTION FOR THE PROJECT.  FUNCTION TO DISPLAY EDIT MENU.  FUNCTION TO DISPLAY EDIT MENU FOR BOOK AND CONTROL  ALL THE FUNCTION IN THE EDIT MENU.  FUNCTION TO DISPLAY EDIT MENU FOR MEMBERS AND CONTROL  ALL THE FUNCTION IN THE EDIT MENU. |
| **BOOK** | THIS CLASS CONTAINS FUNCTIONS RELATED TO BOOKS | | **public :**  void list(void) ;  char \*bookname(int) ;  **protected :**  void add\_new\_book(int, char tname[33], char tauthor[26], float, int, int) ;  void update\_copies(int, int, int);  void modify(int, char[], char[], float) ;  void deletion(void) ;  int book\_found(int) ;  int bookname\_found(char []) ;  int recordno(int) ;  int available(int) ;  char \*authorname(int) ;  float bookprice(int) ;  int no\_of\_copies(int) ;  int bookcodeof(char[]) ;  void display(int) ;  int reccount(void) ;  void delete\_rec(int) ; | THIS FUNCTION DISPLAY THE LIST OF BOOKS.  THIS FUNCTION RETURNS THE BOOK NAME OF THE GIVEN BOOK  CODE.  THIS FUNCTION ADD THE RECORD IN THE BOOK  THIS FUNCTION UPDATE THE RECORD IN THE BOOK FILE FOR THE  GIVEN BOOK CODE FILE  THIS FUNCTION MODIFY THE RECORD IN THE BOOK FILE FOR THE  GIVEN BOOK CODE.  DELETES THE RECORD IN THE FILE.  THIS FUNCTION RETURN 0 IF GIVEN BOOK CODE NOT FOUND.  THIS FUNCTION RETURN 0 IF GIVEN BOOK NAME NOT FOUND.  THIS FUNCTION RETURN RECORD NO. FOR THE BOOK CODE  THIS FUNCTION RETURNS THE AVAILABLE COPIES FOR THE GIVEN  BOOK CODE.  THIS FUNCTION RETURNS THE AUTHOR NAME OF THE GIVEN BOOK  CODE.  THIS FUNCTION RETURNS THE BOOK PRICE OF THE GIVEN BOOK CODE.  THIS FUNCTION RETURNS THE NO. OF COPIES FOR THE GIVEN BOOK CODE.  THIS FUNCTION RETURNS THE BOOK CODE OF THE GIVEN BOOK NAME.  THIS FUNCTION DISPLAY THE RECORD FROM THE BOOK FILE FOR THE GIVEN BOOK CODE  THIS FUNCTION RETURNS THE NO. OF THE RECORDS IN THE BOOK  FILE.  THIS FUNCTION DELETES THE RECORD OF THE GIVEN BOOK CODE. |
| **MEMBER** | | THIS CLASS CONTAINS FUNCTIONS RELATED TO MEMBERS | **public :**  void list(void) ;  **protected :**  void add\_mem(int, int, char [], char [], char[], int, int, int) ;  void modify(int, char[], char[], char[]) ;  int member\_found(int) ;  void update\_book(int, int, int, int, int) ;  char \*membername(int) ;  char \*memberphone(int) ;  char \*memberaddress(int) ;  int recordno(int) ;  int lastcode(void) ;  int issued(int) ;  int fine(int) ;  void display(int) ;  void delete\_rec(int) ; | THIS FUNCTION DISPLAY THE LIST OF THE MEMBERS.  THIS FUNCTION ADD RECORD IN THE FILE FOR THE GIVEN MEMBER CODE.  THIS FUNCTION MODIFY RECORD FOR THE GIVEN MEMBER CODE.  THIS FUNCTION RETURN 0 IF THE GIVEN MEMBER CODE NOT FOUND  THIS FUNCTION UPDATE RECORD FOR THE GIVEN MEMBER CODE.  THIS FUNCTION RETURNS MEMBER NAME OF THE GIVEN MEMBER  CODE.  THIS FUNCTION RETURNS MEMBER PHONE OF THE GIVEN  MEMBER  CODE.  THIS FUNCTION RETURNS MEMBER  ADDRESS OF THE GIVEN MEMBER CODE.  THIS FUNCTION RETURNS RECORD NO. OF THE GIVEN MEMBER CODE.  THIS FUNCTION RETURN THE LAST CODE OF THE MEMBER FILE.  THIS FUNCTION RETURN 0 IF THE MEMBER HAVE NOT ISSUED ANY  BOOK.  THIS FUNCTION CALCULATE AND RETURN FINE FOR THE GIVEN MEMBER CODE.  THIS FUNCTION DISPLAYS THE RECORD FROM THE BOOK FILE FOR THE GIVEN BOOK CODE.  THIS FUNCTION DISPLAY THE RECORD FOR THE GIVEN MEMBER CODE.  THIS FUNCTION DELETE RECORD FOR THE GIVEN MEMBER CODE. |
| **WORKING** | | THIS IS DERIVED FROM CLASS BOOK AND MEMBER AND CONTAINS  FUNCTIONS  FOR WORKING (ISSUE, RETURN, ETC). | public :  void issuebook(void) ;  void returnbook(void) ;  void add\_book(void) ;  void add\_member(void) ;  void modify\_book(void) ;  void modify\_member(void) ;  void delete\_book(void) ;  void delete\_member(void) ; | THIS FUNCTION ISSUES THE BOOK.  THIS FUNCTION RETURNS THE BOOK FOR THE MEMBER.  THIS FUNCTION GIVES DATA TO ADD RECORD IN THE BOOK FILE.  THIS FUNCTION GIVES DATA TO ADD RECORD IN THE MEMBER FILE.  THIS FUNCTION GIVES DATA TO MODIFY THE BOOK RECORD.  THIS FUNCTION GIVES DATA TO MODIFY THE MEMBER RECORD.  THIS FUNCTION GIVES BOOK CODE TO DELETE THE BOOK RECORD.  THIS FUNCTION GIVES MEMBER CODE TO DELETE THE MEMBER RECORD. |
| **DATE** | | THIS CLASS CONTAINS FUNCTIONS RELATED DATE & FINE | void extend\_date(int,int,int,int) ;  int diff(int, int, int, int, int, int) ; | FUNCTION TO EXTEND GIVEN DATE BY 15 DAYS.  THIS FUNCTION RETURN THE DIFFERENCE BETWEEN TWO GIVEN  DATES |

**glossary**

tname- name of the book

tauthor- name of the author

bookcode-code given to each book

copies- no. of copies of each book

price- cost of each book

name- name of the customer

phone- phone no. of the customer

dd- date of issue of book

mm- month of issue of book

yy- year of issue of book

source code

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// PROJECT :LIBRARY MANAGEMENT**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// GROUP 1: AMANDEEP KAUR & HITIKA NARANG**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// HEADER FILES

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <fstream.h>

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include <ctype.h>

#include <conio.h>

#include <dos.h>

**class BOOK**

{

protected:

int bookcode, copies, avail ;

char bookname[26], author[26] ;

float price ;

public :

void list( ) ;

char \* get\_bookname(int) ;

protected :

void add\_new\_book(int, char[], char[], float, int, int) ;

void modify(int, char[], char[], float) ;

void deletion( ) ;

void display(int) ;

int is\_book\_found(char []) ;

void delete\_rec(int) ;

int get\_bookcode(char[]) ;

int get\_available(int) ;

char \* get\_authorname(int) ;

float get\_bookprice(int) ;

int get\_copies(int) ;

void set\_copies(int, int, int) ;

} ;

**class MEMBER**

{

int memcode, bookcode ;

char memname[26], phone[11], address[31] ;

int dd, mm, yy ; //date of returning

public :

void list( ) ;

protected :

void add\_mem(int, int, char[], char[], char[], int, int, int) ;

void modify(int, char[], char[], char[]) ;

void display(int) ;

int issued(int) ;

int fine(int) ;

void delete\_rec(int) ;

int is\_member\_found(char[]) ;

char \* get\_membername(int) ;

int get\_membercode(char[]);

char \* get\_memberphone(int) ;

char \* get\_memberaddress(int) ;

int get\_recordno(int) ;

int get\_lastcode( ) ;

void update\_book(int, int, int, int, int) ;

} ;

**class LIBRARY : public BOOK, public MEMBER**

{

public :

void issue\_book( ) ;

void return\_book( ) ;

void add\_book( ) ;

void add\_member( ) ;

void modify\_book( ) ;

void modify\_member( ) ;

void delete\_book( ) ;

void delete\_member( ) ;

} ;

**class DATE**

{

public :

int day, mon, year ; y

int isLeapYear(int);

void extend\_date(int,int,int,int) ;

int diff(int, int, int, int, int, int) ;

} ;

**void design();**

**void introduction();**

**void main\_menu();**

**void edit\_menu();**

**void edit\_book\_menu();**

**void edit\_member\_menu();**

**int login();**

**void main( )**

{

clrscr();

design() ;

introduction() ;

if (login())

main\_menu();

else

{

cout<<"\n\n Wrong Password .....";

cout<<"\n\n You are NOT authenticated to login...";

getch();

}

}

**int DATE :: isLeapYear (int year)**

{

if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)

return 1;

else

return 0;

}

**void DATE :: extend\_date(int d1, int m1, int y1, int days)**

{

int month[] = {0,31,28,31,30,31,30,31,31,30,31,30,31} ;

if (isLeapYear(y1))

month[2] = 29;

for (int i=1; i<=days; i++)

{

d1++ ;

if (d1 > month[m1])

{

d1 = 1 ;

m1++ ;

}

if (m1 > 12)

{

m1 = 1 ;

y1++ ;

}

}

day = d1 ;

mon = m1 ;

year = y1 ;

}

**int DATE :: diff(int d1, int m1, int y1, int d2, int m2, int y2)**

{

int days = 0 ;

if ((y2<y1) || (y2==y1 && m2<m1) || (y2==y1 && m2==m1 && d2<d1))

return days ;

int month[] = {0,31,28,31,30,31,30,31,31,30,31,30,31} ;

if (isLeapYear(y1))

month[2] = 29;

while (d1 != d2 || m1 != m2 || y1 != y2)

{

days++ ;

d1++ ;

if ((d1 > month[m1]))

{

d1 = 1 ;

m1++ ;

}

if (m1 > 12)

{

m1 = 1 ;

y1++ ;

}

}

return days ;

}

**void design( )**

{

cout<<"\n\t L i bbbb rrrr A rrrr y y O ";

cout<<"\n\t L I B B R R A A R R y y ";

cout<<"\n\t L I bbbb rrrr AaaaA rrrr y y ";

cout<<"\n\t L I B B R R A A R R y ";

cout<<"\n\t LLLLLLL I bbbb R R A A R R y O \n\t ";

for (int i=0;i<55;i++)

{ cout<<"\_";delay(10); }

cout<<"\n\n\n";

cout<<"\n\_\_\_ \_ \_\_\_ \_ \_ \_ \_ \_\_ \_ \_ \_ o \_\_ ";

cout<<"\n | |\_||\_ | |\_ |\\/||\_) | |\_ | | |\_ | |\_ |\_| |\_) |\\ | | |\\ | | \_ ";

cout<<"\n | | ||\_ | |\_ | || |\_\_ |\_ |\_| | |\_\_ |\_ | | | \\ | \\| | | \\| |\_\_| o o ";

for(int j=0;j<75;j++)

{ cout<<"x";delay(70); }

cout<<"\n\n\n\t\t\t PRESS ANY KEY TO CONTINUE..............";

getch();

}

**void main\_menu( )**

{

LIBRARY L;

BOOK B;

MEMBER M;

char ch ;

while (1)

{

clrscr() ;

gotoxy(29,6) ; cout <<"B O O K L I B R A R Y" ;

gotoxy(29,7) ; cout <<"~~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(30,10) ;cout <<"1. INTRODUCTION" ;

gotoxy(30,11) ;cout <<"2. ADD NEW BOOK" ;

gotoxy(30,12) ;cout <<"3. ADD NEW MEMBER" ;

gotoxy(30,13) ;cout <<"4. ISSUE BOOK" ;

gotoxy(30,14) ;cout <<"5. RETURN BOOK" ;

gotoxy(30,15) ;cout <<"6. LIST OF BOOKS" ;

gotoxy(30,16) ;cout <<"7. LIST OF MEMBERS" ;

gotoxy(30,17) ;cout <<"8. EDIT" ;

gotoxy(30,18) ;cout <<"0. QUIT" ;

gotoxy(30,20) ;cout <<"Enter Your Choice [0-8] : " ;

ch = getche() ;

if (ch == '0')

break;

else if (ch == '1')

introduction() ;

else if (ch == '2')

L.add\_book() ;

else if (ch == '3')

L.add\_member() ;

else if (ch == '4')

L.issue\_book() ;

else if (ch == '5')

L.return\_book() ;

else if (ch == '6')

B.list() ;

else if (ch == '7')

M.list() ;

else if (ch == '8')

edit\_menu() ;

else

{

gotoxy(30, 22);

cout << "Invalid Choice. Press any key to continue...";

getch();

}

}

}

**void edit\_menu( )**

{

char ch ;

while(1)

{ clrscr();

gotoxy(31,9) ; cout <<"E D I T M E N U";

gotoxy(31,10) ;cout <<"~~~~~~~~~~~~~~~~" ;

gotoxy(34,13) ;cout <<"1. BOOKS" ;

gotoxy(34,14) ;cout <<"2. MEMBERS" ;

gotoxy(34,15) ;cout <<"0. EXIT" ;

gotoxy(31,17) ;cout <<"Enter Your Choice [0-2] : " ;

ch = getche() ;

if (ch == '0')

break;

else if (ch == '1')

edit\_book\_menu() ;

else if (ch == '2')

edit\_member\_menu() ;

else

{ gotoxy(31, 19);

cout << "Invalid Choice. Press any key to continue...";

getch();

}

};

}

**void edit\_book\_menu( )**

{

LIBRARY L ;

char ch ;

while(1)

{ clrscr() ;

gotoxy(31,9) ; cout <<"E D I T B O O K S";

gotoxy(31,10) ;cout <<"~~~~~~~~~~~~~~~~~~" ;

gotoxy(34,13) ;cout <<"1. MODIFY" ;

gotoxy(34,14) ;cout <<"2. DELETE" ;

gotoxy(34,15) ;cout <<"0. EXIT" ;

gotoxy(31,17) ;cout <<"Enter Your Choice [0-2] : " ;

ch = getche() ;

if (ch == '0')

break;

else if (ch == '1')

L.modify\_book() ;

else if (ch == '2')

L.delete\_book() ;

else

{ gotoxy(31, 19);

cout << "Invalid Choice. Press any key to continue...";

getch();

}

};

}

**void edit\_member\_menu( )**

{

LIBRARY L;

char ch ;

while(1)

{ clrscr() ;

gotoxy(29,9) ; cout <<"E D I T M E M B E R S";

gotoxy(29,10) ;cout <<"~~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(34,13) ;cout <<"1. MODIFY" ;

gotoxy(34,14) ;cout <<"2. DELETE" ;

gotoxy(34,15) ;cout <<"0. EXIT" ;

gotoxy(29,17) ;cout <<"Enter Your Choice [0-2] : " ;

ch = getche() ;

if (ch == '0')

break;

else if (ch == '1')

L.modify\_member() ;

else if (ch == '2')

L.delete\_member() ;

else

{ gotoxy(29, 19);

cout << "Invalid choice. Press any key to continue...";

getch();

}

};

}

**void introduction( )**

{

clrscr() ;

gotoxy(31,5) ;

cout <<" WELCOME TO THE PROJECT " ;

textcolor(BLACK+BLINK) ; textbackground(CYAN) ;

gotoxy(33,7) ;

cprintf(" LIBRARY MANAGEMENT ") ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

gotoxy(15,10) ;

cout <<"This project has facility of maintaining book records" ;

gotoxy(15,12) ;

cout <<"of BOOKS and MEMBERS." ;

gotoxy(15,14) ;

cout <<"Only one book is issued to a member at a time." ;

gotoxy(15,16) ;

cout <<"If he/she does not return the book within 15 days," ;

gotoxy(15,18) ;

cout <<"he/she is liable to pay fine of Rs.2/- per day." ;

gotoxy(15,21) ;

textcolor(LIGHTGRAY+BLINK) ;

cprintf("Press any key to continue") ;

textcolor(LIGHTGRAY) ;

getch() ;

}

**int BOOK :: is\_book\_found(char code\_name[])**

{

fstream file ("BOOK.DAT", ios::in) ;

int found=0, bcode = atoi (code\_name);

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode || strcmpi(bookname,code\_name)==0)

{ found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

**int BOOK :: get\_available(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

int bavail=0 ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode)

{ bavail = avail ;

break ;

}

}

file.close() ;

return bavail ;

}

**int BOOK :: get\_copies(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

int bcopies ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode)

{ bcopies = copies ;

break ;

}

}

file.close() ;

return bcopies ;

}

**char \*BOOK :: get\_bookname(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

char bname[31] ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode)

{ strcpy(bname,bookname) ;

break ;

}

}

file.close() ;

return bname ;

}

**char \*BOOK :: get\_authorname(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

char bauthor[26] ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode)

{ strcpy(bauthor,author) ;

break ;

}

}

file.close() ;

return bauthor ;

}

**float BOOK :: get\_bookprice(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

float bprice ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode)

{ bprice = price ;

break ;

}

}

file.close() ;

return bprice ;

}

**int BOOK :: get\_bookcode(char bname[])**

{

fstream file ("BOOK.DAT", ios::in) ;

int bcode ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (strcmpi(bookname,bname)==0)

{ bcode = bookcode ;

break ;

}

}

file.close() ;

return bcode ;

}

**void BOOK :: delete\_rec(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

fstream temp ("TEMP.DAT", ios::out) ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if ( bookcode != bcode )

temp.write((char \*) this, sizeof(BOOK)) ;

}

file.close() ;

temp.close() ;

remove("BOOK.DAT");

rename("TEMP.DAT","BOOK.DAT");

}

**void BOOK :: add\_new\_book(int bcode,char bname[], char bauthor[], float bprice, int bcopies, int bavail)**

{

fstream file ("BOOK.DAT", ios::app) ;

bookcode = bcode ;

strcpy(bookname,bname) ;

strcpy(author,bauthor) ;

price = bprice ;

copies = bcopies ;

avail = bavail ;

file.write((char \*) this, sizeof(BOOK)) ;

file.close() ;

}

**void BOOK :: set\_copies(int bcode, int bcopies, int bavail)**

{

fstream file ("BOOK.DAT", ios::in) ;

fstream temp ("TEMP.DAT", ios::out) ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if ( bookcode == bcode )

{ copies = bcopies ;

avail = bavail ;

}

temp.write((char \*) this, sizeof(BOOK)) ;

}

file.close() ;

temp.close() ;

remove("BOOK.DAT");

rename("TEMP.DAT","BOOK.DAT");

}

**void BOOK :: modify(int bcode, char bname[], char bauthor[], float bprice)**

{

fstream file ("BOOK.DAT", ios::in) ;

fstream temp ("temp.dat", ios::out) ;

while (file.read((char \*) this, sizeof(BOOK)) )

{ if ( bookcode == bcode )

{ strcpy(bookname,bname) ;

strcpy(author,bauthor) ;

price = bprice ;

}

temp.write((char \*) this, sizeof(BOOK)) ;

}

file.close() ;

temp.close() ;

remove("BOOK.DAT");

rename("TEMP.DAT","BOOK.DAT");

}

**void BOOK :: display(int bcode)**

{

fstream file ("BOOK.DAT", ios::in) ;

while (file.read((char \*) this, sizeof(BOOK)))

{ if (bookcode == bcode)

{ gotoxy(5,5) ; cout <<"Book Code : " <<bookcode ;

gotoxy(5,7) ; cout <<"Book Name : " <<bookname ;

gotoxy(5,8) ; cout <<"Author Name : " <<author ;

gotoxy(5,9) ; cout <<"Price : Rs." <<price ;

gotoxy(5,10) ;cout <<"Copies : " <<copies ;

gotoxy(5,11) ;cout <<"Available : " <<avail ;

break ;

}

}

file.close() ;

}

**void BOOK :: list( )**

{

clrscr() ;

int row = 6, found=0;

char ch ;

gotoxy(32,2) ; cout <<" LIST OF BOOKS" ;

gotoxy(32,3) ; cout <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" ;

gotoxy(1,4) ;

cout <<"CODE BOOK NAME AUTHOR PRICE COPIES AVAILABLE" ;

gotoxy(1,5) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

fstream file ("BOOK.DAT", ios::in) ;

while (file.read((char \*) this, sizeof(BOOK)))

{ found = 1 ;

gotoxy(1,row) ; cout << bookcode ;

gotoxy(7,row) ; cout << bookname ;

gotoxy(35,row) ; cout << author ;

gotoxy(56,row) ; cout << price ;

gotoxy(65,row) ; cout << copies ;

gotoxy(75,row) ; cout<<avail;

row = row + 2 ;

}

file.close () ;

if (!found)

{ gotoxy(1,row) ; cout <<"Records Not found" ;

}

gotoxy(1, 24) ; cout <<"Press any key to continue..." ;

getch() ;

}

**int MEMBER :: is\_member\_found(char code\_name[])**

{

fstream file ("MEMBER.DAT", ios::in) ;

int found=0 ;

int mcode = atoi(code\_name);

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode || strcmpi(memname, code\_name)==0)

{ found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

**int MEMBER :: issued(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

int bcode ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode)

{ bcode = bookcode ;

break ;

}

}

file.close() ;

return bcode ;

}

**int MEMBER :: fine(int mcode)**

{

DATE D ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

fstream file ("MEMBER.DAT", ios::in) ;

int days, mfine ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode)

{ days = D.diff(dd,mm,yy,d1,m1,y1) ;

mfine = days \* 2 ;

break ;

}

}

file.close() ;

return mfine ;

}

**int MEMBER :: get\_lastcode( )**

{

fstream file ("MEMBER.DAT", ios::in) ;

int mcode=0 ;

while (file.read((char \*) this, sizeof(MEMBER)))

mcode = memcode ;

file.close() ;

return mcode ;

}

**char \*MEMBER :: get\_membername(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

char mname[26] ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode)

{ strcpy(mname,memname) ;

break ;

}

}

file.close() ;

return mname ;

}

**int MEMBER :: get\_membercode(char mname[])**

{

fstream file ("MEMBER.DAT", ios::in) ;

int mcode;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (strcmpi(memname, mname)==0)

{ mcode = memcode;

break ;

}

}

file.close() ;

return mcode ;

}

**char \*MEMBER :: get\_memberphone(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

char mphone[11] ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode)

{ strcpy(mphone,phone) ;

break ;

}

}

file.close() ;

return mphone ;

}

**char \*MEMBER :: get\_memberaddress(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

char maddress[31] ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode)

{ strcpy(maddress,address) ;

break ;

}

}

file.close() ;

return maddress ;

}

**int MEMBER :: get\_recordno(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

int count=0 ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ count++ ;

if (memcode == mcode)

break ;

}

file.close() ;

return count ;

}

**void MEMBER :: delete\_rec(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

fstream temp ("TEMP.DAT", ios::out) ;

while ( file.read((char \*) this, sizeof(MEMBER)))

{ if ( memcode != mcode )

temp.write((char \*) this, sizeof(MEMBER)) ;

}

file.close() ;

temp.close() ;

remove("MEMBER.DAT");

rename("TEMP.DAT","MEMBER.DAT");

}

**void MEMBER :: update\_book(int mcode, int bcode, int d1, int m1, int y1)**

{

fstream file ("MEMBER.DAT", ios::in) ;

fstream temp ("TEMP.DAT", ios::out) ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if ( memcode == mcode )

{ bookcode = bcode ;

dd = d1 ;

mm = m1 ;

yy = y1 ;

}

temp.write((char \*) this, sizeof(MEMBER)) ;

}

file.close() ;

temp.close() ;

remove("MEMBER.DAT") ;

rename("TEMP.DAT", "MEMBER.DAT") ;

}

**void MEMBER :: modify(int mcode, char mname[], char mphone[], char maddress[])**

{

fstream file ("MEMBER.DAT", ios::in) ;

fstream temp ("TEMP.DAT", ios::out) ;

while ( file.read((char \*) this, sizeof(MEMBER)))

{ if ( memcode == mcode )

{ strcpy(memname,mname) ;

strcpy(phone,mphone) ;

strcpy(address,maddress) ;

}

temp.write((char \*) this, sizeof(MEMBER)) ;

}

file.close() ;

temp.close() ;

remove("MEMBER.DAT") ;

rename("TEMP.DAT", "MEMBER.DAT") ;

}

**void MEMBER :: add\_mem(int mcode, int bcode, char mname[], char maddress[], char mphone[], int d1, int m1, int y1)**

{

fstream file ("MEMBER.DAT", ios::app) ;

memcode = mcode ;

bookcode = bcode ;

strcpy(memname,mname) ;

strcpy(address,maddress) ;

strcpy(phone,mphone) ;

dd = d1 ;

mm = m1 ;

yy = y1 ;

file.write((char \*) this, sizeof(MEMBER)) ;

file.close() ;

}

**void MEMBER :: display(int mcode)**

{

fstream file ("MEMBER.DAT", ios::in) ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ if (memcode == mcode)

{ gotoxy(5,3) ;cout <<"Member Code # " <<memcode ;

gotoxy(5,4) ;cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(5,6) ;cout <<"Name : " <<memname ;

gotoxy(5,7) ;cout <<"Phone : " <<phone ;

gotoxy(5,8) ;cout <<"Address : " <<address ;

break ;

}

}

file.close() ;

}

**void MEMBER :: list( )**

{

clrscr() ;

BOOK B ;

int row = 6, found=0;

gotoxy(32,2) ;cout <<"LIST OF MEMBERS" ;

gotoxy(31,3) ;cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ; cout <<"CODE NAME PHONE BOOK NAME (CODE) RETURN DATE" ;

gotoxy(1,5) ; cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

fstream file ("MEMBER.DAT", ios::in) ;

while (file.read((char \*) this, sizeof(MEMBER)))

{ found = 1 ;

gotoxy(2,row) ; cout <<memcode ;

gotoxy(7,row) ; cout <<memname ;

gotoxy(25,row) ; cout <<phone ;

gotoxy(40,row) ;

if (bookcode == 0)

{ cout <<"Not Issued" ;

gotoxy(68,row); cout << "xx/xx/xxxx";

}

else

{ cout <<B.get\_bookname(bookcode)<<" ("<<bookcode<<")";

gotoxy(68,row) ; cout << dd << "/" << mm << "/" << yy ;

}

row += 2 ;

}

if (!found)

{ gotoxy(5,10) ; cout <<"\nRecords Not Found" ;

}

gotoxy(1,24) ; cout <<"Press any key to continue..." ;

getche() ;

file.close () ;

}

**void LIBRARY :: add\_book( )**

{

int bcode, bcopies, bavail, found ;

char ch, strbcode[26], bname[26], bauthor[26], strprice[10], strcopies[5] ;

float bprice ;

do

{ found=0 ;

clrscr() ;

gotoxy(29,3) ; cout <<"ADDITION OF THE BOOKS" ;

gotoxy(29,4) ; cout <<"~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,5) ; cout <<"Code No: " ;

gets(strbcode) ;

bcode = atoi(strbcode) ;

if (strlen(strbcode)==0)

return;

if (bcode==0)

{ gotoxy(5,7); cout << "Invalid Code. Press any key to continue...";

getch();

return ;

}

gotoxy(5,8) ; cout << "Book Name : ";

gotoxy(5,10) ; cout << "Author Name : ";

gotoxy(5,12); cout << "Price : Rs.";

gotoxy(5,14); cout << "Copies : (available)";

gotoxy(5,16); cout << "Copies : " ;

if (is\_book\_found(strbcode))

{ found = 1;

gotoxy(19,8) ; cout << get\_bookname(bcode);

gotoxy(19,10) ; cout << get\_authorname(bcode);

gotoxy(19,12); cout << get\_bookprice(bcode);

gotoxy(19,14); cout << get\_copies(bcode) << " (available)";

gotoxy(19,16);

gets(strcopies) ;

bcopies = atoi(strcopies) ;

bavail = get\_available(bcode) + bcopies ;

bcopies = get\_copies(bcode) + bcopies ;

}

else

{ gotoxy(19,8); gets(bname) ;

strupr(bname) ;

gotoxy(19,10); gets(bauthor) ;

strupr(bauthor) ;

gotoxy(22,12); gets(strprice) ;

bprice = atof(strprice) ;

gotoxy(5,14); cout << "Copies : 0 (available)";

gotoxy(19,16); gets(strcopies) ;

bcopies = atoi(strcopies) ;

bavail = bcopies ;

}

gotoxy(5,18) ; cout <<"Want to save (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == 'Y')

{ if (found)

set\_copies(bcode,bcopies,bavail) ;

else

add\_new\_book(bcode,bname,bauthor,bprice,bcopies,bavail) ;

}

gotoxy(5,20) ; cout <<"Want to add more (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

} while (ch == 'Y') ;

}

**void LIBRARY :: add\_member( )**

{

char ch, mname[26], mphone[11], maddress[31] ;

int bcode, d1, m1, y1 ;

int mcode = get\_lastcode()+1 ;

do

{ clrscr() ;

gotoxy(28,3) ; cout <<"ADDITION OF THE MEMBERS" ;

gotoxy(28,4) ; cout <<"~~~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,7) ; cout <<"Member Code # " <<mcode ;

gotoxy(5,8) ; cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(5,10) ; cout <<"Name : " ;

gotoxy(15,10) ;gets(mname) ;

strupr(mname) ;

gotoxy(5,12) ; cout <<"Phone : " ;

gotoxy(15,12) ;gets(mphone) ;

gotoxy(5,14) ; cout <<"Address : ";

gotoxy(15,14) ;gets(maddress) ;

strupr(maddress) ;

gotoxy(5,17) ; cout <<"Want to save (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == 'Y')

{ bcode = 0 ;

d1 = 0 ;

m1 = 0 ;

y1 = 0 ;

add\_mem(mcode,bcode,mname,maddress,mphone,d1,m1,y1) ;

mcode++ ;

}

gotoxy(5,19) ; cout <<"Want to add more (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

} while (ch == 'Y') ;

}

**void LIBRARY :: issue\_book( )**

{

BOOK B ;

MEMBER M ;

DATE D ;

char code\_name[26], ch ;

int bcode, mcode ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

clrscr() ;

gotoxy(5,2) ; cout <<"Date : " <<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,5) ; cout <<"Enter Code or Name of the Book : " ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name) == 0)

return ;

if (!is\_book\_found(code\_name))

{ gotoxy(5,10) ;

cout <<"Record Not Found. Press any key to continue..." ;

getch() ;

return ;

}

bcode = atoi(code\_name) ;

if (bcode == 0)

bcode = get\_bookcode(code\_name);

if (get\_available(bcode) == 0)

{ gotoxy(5,10) ;

cout <<"Sorry! Book (" <<get\_bookname(bcode) <<") is not available." ;

gotoxy(5,12) ;

cout <<"Press any key to continue ..." ;

getch() ;

return ;

}

clrscr() ;

gotoxy(5,2) ; cout <<"Date : " <<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,5) ; cout << "Book Name : " << get\_bookname(bcode) ;

gotoxy(5,7) ; cout << "Author : " << get\_authorname(bcode);

gotoxy(5,9) ; cout << "Price : Rs." << get\_bookprice(bcode);

gotoxy(5,11); cout << "Copies : " << get\_available(bcode) << " (available)";

gotoxy(5,13) ; cout <<"Enter Code / Name of the Member: " ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name) == 0)

return ;

if (!is\_member\_found(code\_name))

{ gotoxy(5,15) ;

cout <<"Record Not Found. Press any key to continue..." ;

getch() ;

return ;

}

mcode = atoi(code\_name);

if (mcode == 0)

mcode = get\_membercode(code\_name);

if (issued(mcode))

{ gotoxy(5,15) ;

cout <<"A book is already issued to " << get\_membername(mcode);

gotoxy(5,16) ;

cout <<"Press any key to continue..." ;

getch() ;

return ;

}

int bcopies, bavail ;

bcopies = get\_copies(bcode) ;

bavail = get\_available(bcode) - 1 ;

set\_copies(bcode,bcopies,bavail) ;

D.extend\_date(d1,m1,y1,15) ;

d1 = D.day ;

m1 = D.mon ;

y1 = D.year ;

update\_book(mcode,bcode,d1,m1,y1) ;

gotoxy(5,15) ;

cout <<"Book Issued to " <<get\_membername(mcode) << " Successfully";

gotoxy(5,17) ;

cout <<"Date of Return : " <<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(5,19) ;

cout <<"Press any key to continue..." ;

getch() ;

}

**void LIBRARY :: return\_book( )**

{

MEMBER M ;

char code\_name[26], ch ;

int mcode;

clrscr() ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

struct date d;

getdate(&d);

int d1 = d.da\_day;

int m1 = d.da\_mon;

int y1 = d.da\_year;

gotoxy(5,2) ; cout <<"Date : " <<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(5,7) ; cout <<"Enter Code or Name of the Member : " ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name)==0)

return ;

if (!is\_member\_found(code\_name))

{ gotoxy(5,13) ;

cout <<"Record Not Found / Invalid Member Code" ;

gotoxy(5,14) ;

cout <<"Press any key to continue..." ;

getch() ;

return ;

}

mcode = atoi(code\_name);

if (mcode == 0)

mcode = get\_membercode(code\_name);

int bcode = issued(mcode) ;

if (bcode == 0)

{ gotoxy(5,13) ; cout <<"Member has no book to return" ;

gotoxy(5,14) ; cout <<"Press any key to continue..." ;

getch() ;

return ;

}

int bcopies, bavail ;

gotoxy(5,13) ; cout <<"Book Code : " <<bcode ;

gotoxy(5,14) ; cout <<"Book Name : " <<get\_bookname(bcode) ;

bcopies = get\_copies(bcode) ;

bavail = get\_available(bcode) + 1 ;

int fineAmt = fine(mcode) ;

if (fineAmt != 0)

{ gotoxy(5,16) ;

cout <<"You have to pay a fine of Rs." << fineAmt ;

gotoxy(5,17) ;

cout <<"Please do not delay the Return of Book again" ;

}

set\_copies(bcode,bcopies,bavail) ;

update\_book(mcode,0,0,0,0) ;

gotoxy(5,19) ;

cout <<"Book Received Successfully. Press any key to continue..." ;

getch() ;

}

**void LIBRARY :: modify\_book( )**

{

BOOK B ;

char code\_name[26], bname[33], bauthor[26], strprice[10], ch ;

int bcode ;

float bprice ;

clrscr() ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,5) ; cout <<"Enter Code or Name of the Book : " ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name) == 0)

return ;

if (!is\_book\_found(code\_name))

{ gotoxy(5,10) ;

cout <<"Record Not Found. Press any key to continue..." ;

getch() ;

return ;

}

clrscr() ;

bcode = atoi(code\_name) ;

if (bcode == 0)

bcode = get\_bookcode(code\_name);

BOOK::display(bcode) ;

gotoxy(5,16) ; cout <<"Book Name : " ;

gotoxy(5,17) ; cout <<"Author Name : " ;

gotoxy(5,18) ; cout <<"Price : Rs." ;

gotoxy(5,25) ; cout <<"Enter Name of book or <ENTER> for no change" ;

gotoxy(19,16) ;gets(bname) ;

strupr(bname) ;

if (strlen(bname) == 0)

strcpy(bname,get\_bookname(bcode)) ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Author's Name or <ENTER> for no change" ;

gotoxy(19,17) ; clreol() ;

gets(bauthor) ;

strupr(bauthor) ;

if (strlen(bauthor) == 0)

strcpy(bauthor,get\_authorname(bcode)) ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Price or <ENTER> for no change" ;

gotoxy(22,18) ; clreol() ;

gets(strprice) ;

gotoxy(5,25); clreol();

if (strlen(strprice) == 0)

bprice = get\_bookprice(bcode) ;

else

bprice = atof(strprice) ;

gotoxy(5,20) ; clreol() ;

cout <<"Want to save changes (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == 'Y')

{

BOOK::modify(bcode,bname,bauthor,bprice) ;

gotoxy(5,23) ;

cout <<"\nRecord Modified Successfully. Press any key to continue..." ;

getch() ;

} }

**void LIBRARY :: modify\_member( )**

{

clrscr();

MEMBER M ;

char code\_name[26], mname[26], mphone[11], maddress[31], ch ;

int mcode ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,7) ; cout <<"Enter Code or Name of the Member : " ;

gotoxy(40,7) ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name)==0)

return ;

if (!is\_member\_found(code\_name))

{ gotoxy(5,13) ;

cout <<"Record Not Found. Press any key to continue..." ;

getch() ;

return ;

}

clrscr() ;

mcode = atoi(code\_name) ;

if (mcode == 0)

mcode = get\_membercode(code\_name);

MEMBER::display(mcode) ;

gotoxy(5,13) ; cout <<"Name : " ;

gotoxy(5,14) ; cout <<"Phone : " ;

gotoxy(5,15) ; cout <<"Address : " ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Name of Member or <ENTER> for no change" ;

gotoxy(15,13) ; gets(mname) ;

strupr(mname) ;

if (strlen(mname) == 0)

strcpy(mname,get\_membername(mcode)) ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Phone No. of Member or <ENTER> for no change" ;

gotoxy(15,14) ; gets(mphone) ;

if (strlen(mphone) == 0)

strcpy(mphone, get\_memberphone(mcode)) ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Address of Member or <ENTER> for no change" ;

gotoxy(15,15) ; gets(maddress) ;

strupr(maddress);

gotoxy(5,25); clreol();

if (strlen(maddress) == 0)

strcpy(maddress, get\_memberaddress(mcode)) ;

gotoxy(5,18) ; clreol() ;

cout <<"Want to save changes (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == 'Y')

{

MEMBER::modify(mcode,mname,mphone,maddress) ;

gotoxy(5,23) ;

cout <<"Record Modified Successfully. Press any key to continue..." ;

getch() ;

}

}

**void LIBRARY :: delete\_book( )**

{

BOOK B ;

char code\_name[26], bname[26], bauthor[26], ch ;

int bcode ;

clrscr() ;

gotoxy(72,1) ;cout <<"<0>=Exit" ;

gotoxy(5,5) ; cout <<"Enter Code or Name of the Book : " ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name) == 0)

return ;

if (!is\_book\_found(code\_name))

{ gotoxy(5,10) ;

cout <<"Record Not Found. Press any key to continue..." ;

getch() ;

return ;

}

clrscr() ;

bcode = atoi(code\_name) ;

if (bcode == 0)

bcode = get\_bookcode(code\_name);

BOOK::display(bcode) ;

gotoxy(5,13) ; clreol() ;

cout <<"Want to delete this record (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == 'Y')

{

int bavail, bcopies ;

bavail = get\_available(bcode) ;

bcopies = get\_copies(bcode) ;

if (bavail != bcopies)

{ gotoxy(5,15) ;

cout <<"Record cannot be deleted. This book is issued." ;

}

else

{ BOOK::delete\_rec(bcode) ;

gotoxy(5,15) ; cout <<"Record Deleted.";

}

}

gotoxy(5,17) ; cout <<"Press any key to continue..." ;

getch() ;

}

**void LIBRARY :: delete\_member( )**

{ MEMBER M ;

char code\_name[26], mname[26], mphone[11], maddress[31], ch ;

int mcode ;

clrscr() ;

gotoxy(72,1) ; cout <<"<0>=Exit" ;

gotoxy(5,7) ; cout <<"Enter Code or Name of the Member : " ;

gets(code\_name) ;

if (code\_name[0] == '0' || strlen(code\_name)==0)

return ;

if (!is\_member\_found(code\_name))

{ gotoxy(5,13) ;

cout <<"Record Not Found. Press any key to continue..." ;

ch = getch() ;

return ;

}

clrscr() ;

mcode = atoi(code\_name) ;

if (mcode == 0)

mcode = get\_membercode(code\_name);

MEMBER::display(mcode) ;

gotoxy(5,10) ; clreol() ;

cout <<"Want to Delete this record (Y/N) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == 'N')

return ;

if (issued(mcode))

{

gotoxy(5,15) ;

cout <<"Record cannot be deleted because a book is issued to him." ;

}

else

{ MEMBER::delete\_rec(mcode) ;

gotoxy(5,23) ; cout <<"Record Deleted Successfully." ;

}

gotoxy(5,17) ; cout << "Press any key to continue...";

getch() ;

}

**int login()**

{ clrscr();

char mypwd[25]="library",str[25]="";

char ch;

int i=0;

gotoxy(25,9) ; cout<<" Enter The Password : ";

while(ch!='\r')

{ ch=getch();

cout<<"\*";

str[i]=ch;

i++;

}

str[i-1]='\0';

if(strcmp(mypwd,str)!=0)

return 0;

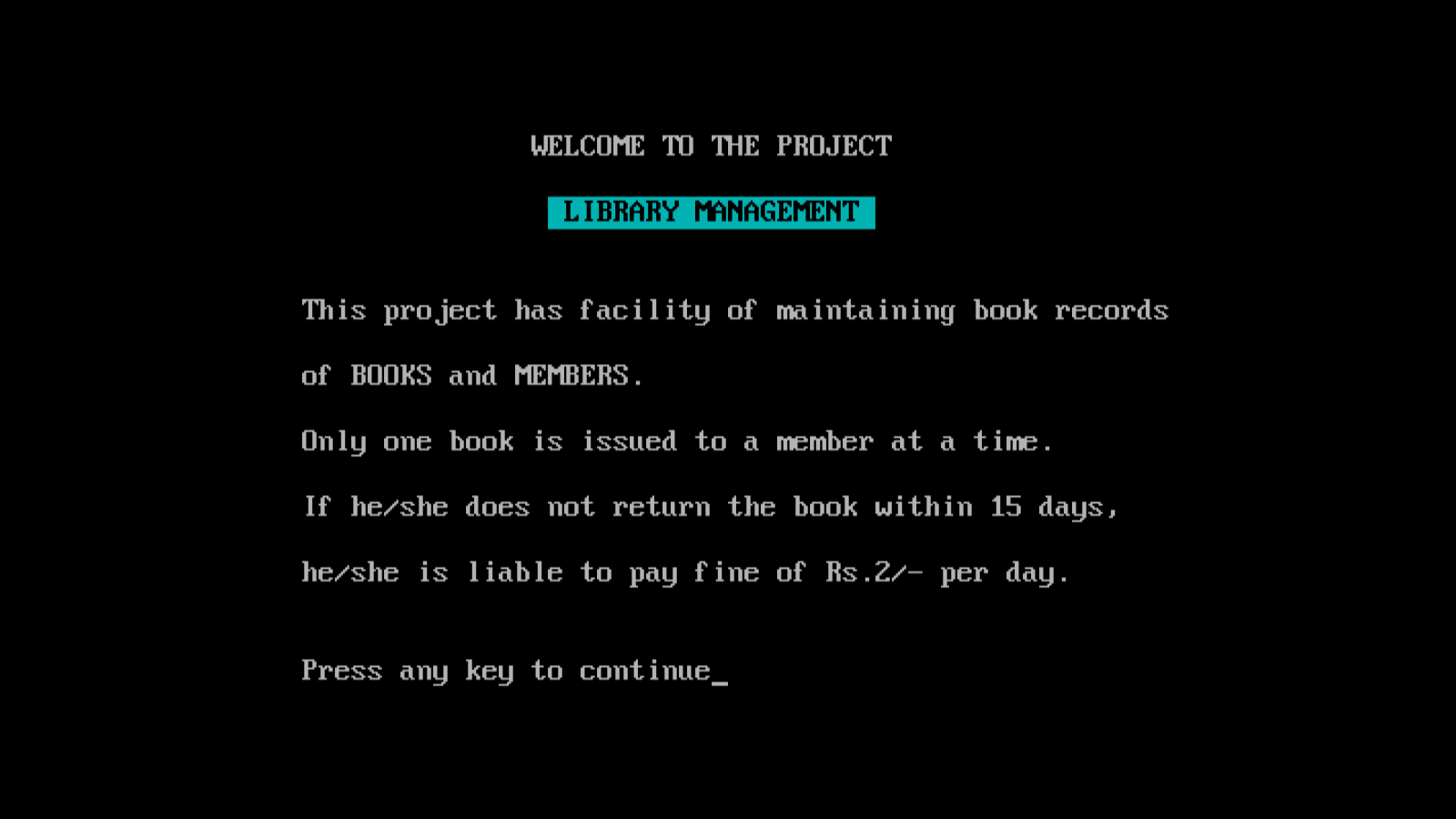
else

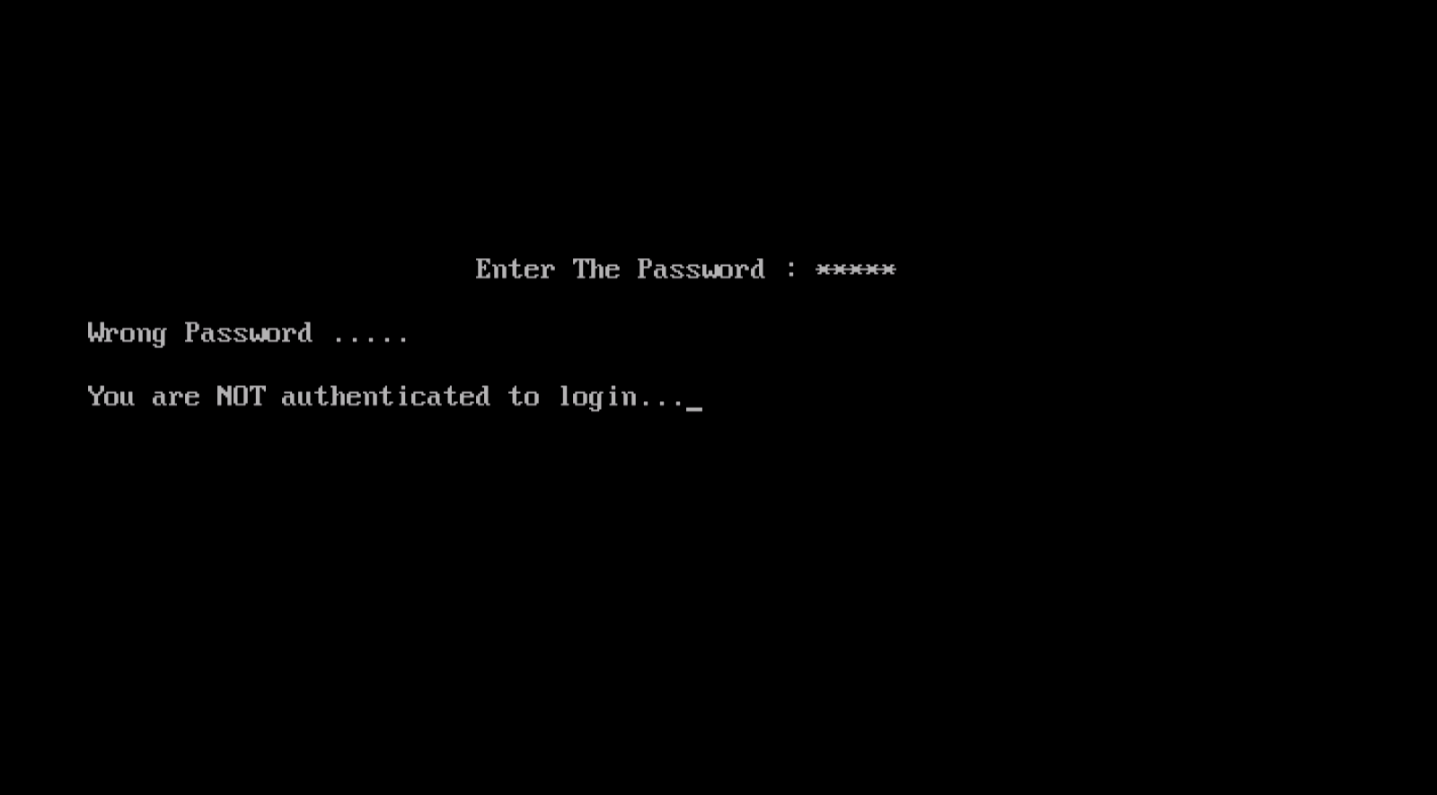
return 1; }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

O**utputs..**

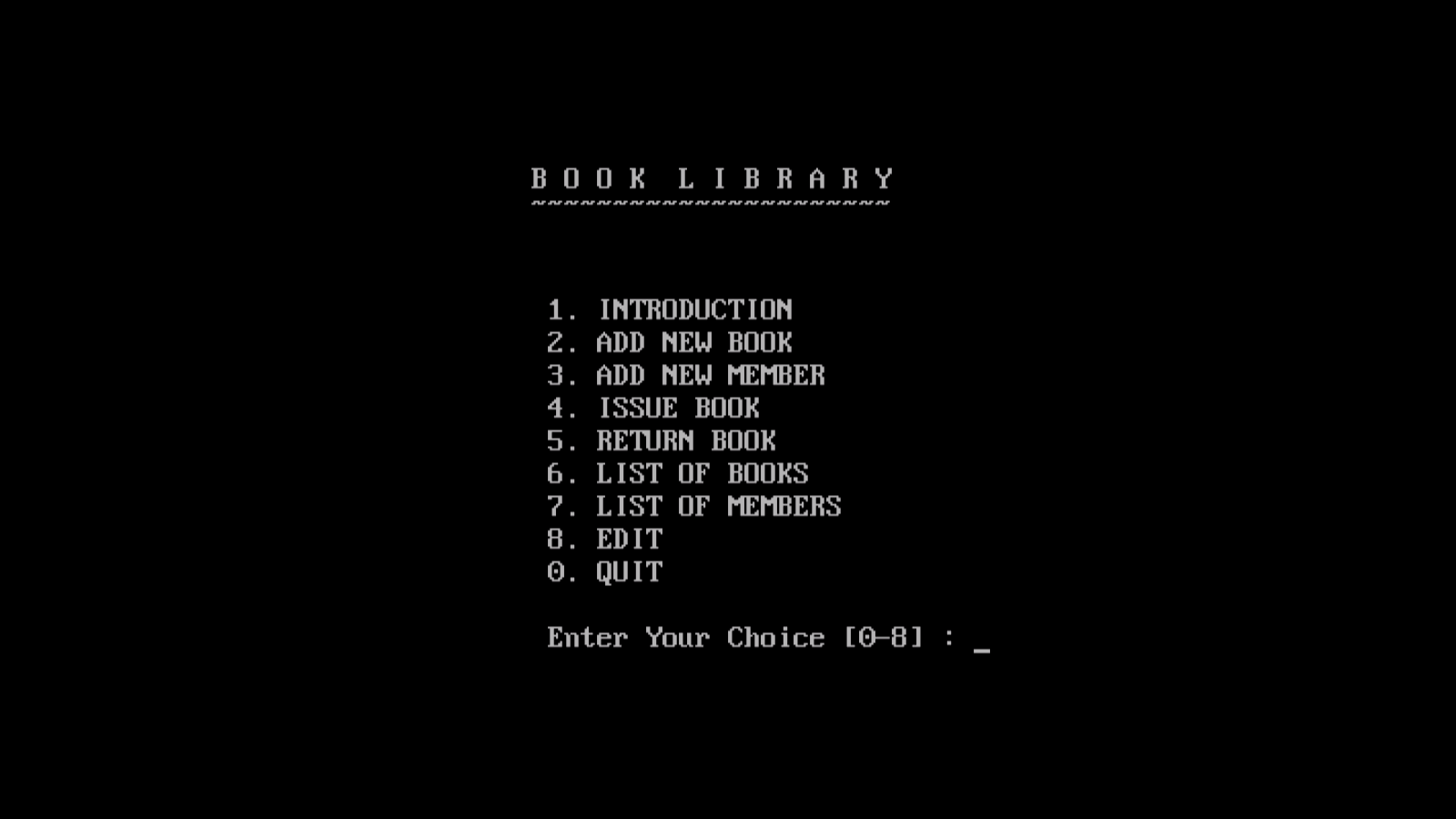


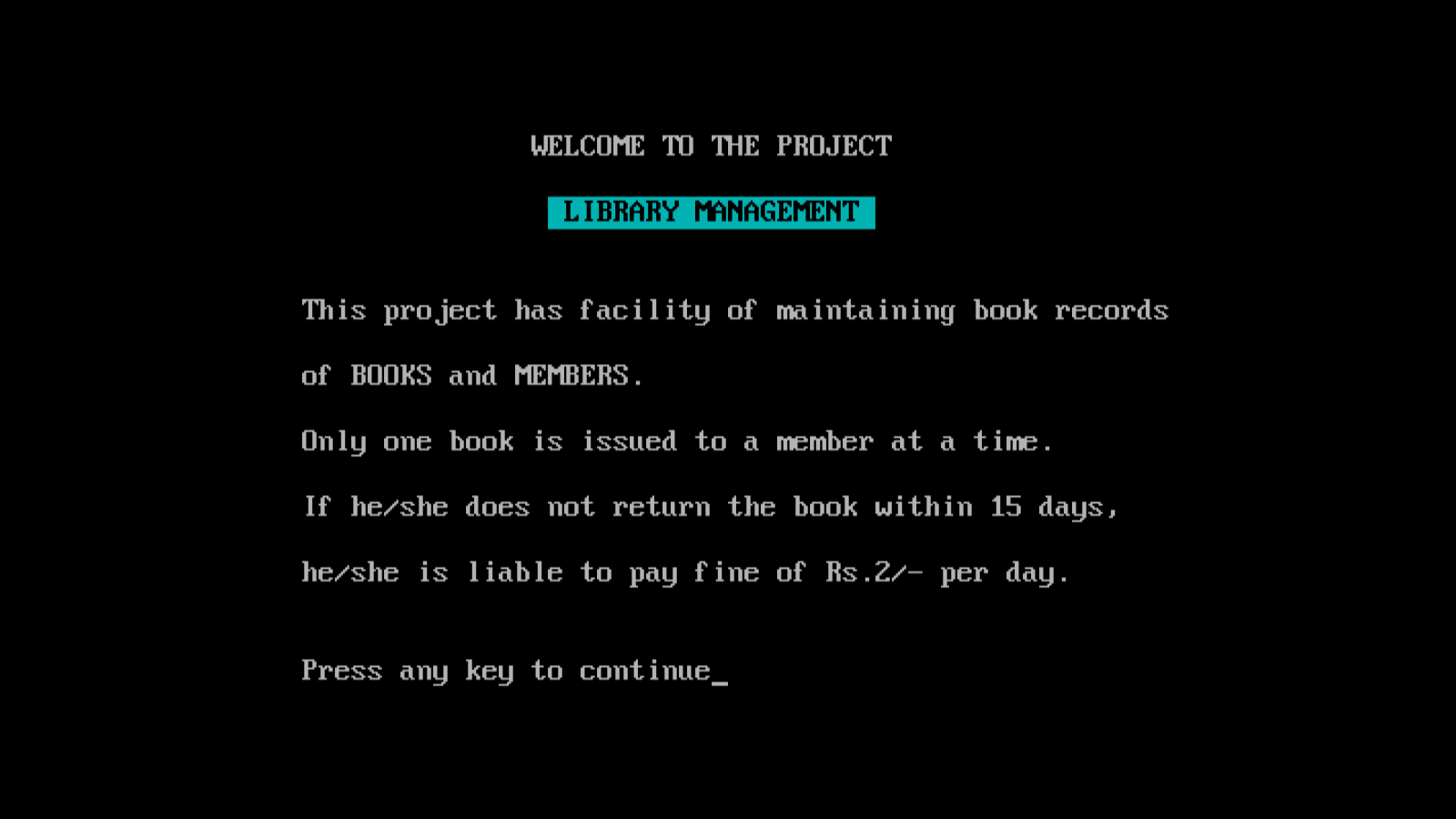


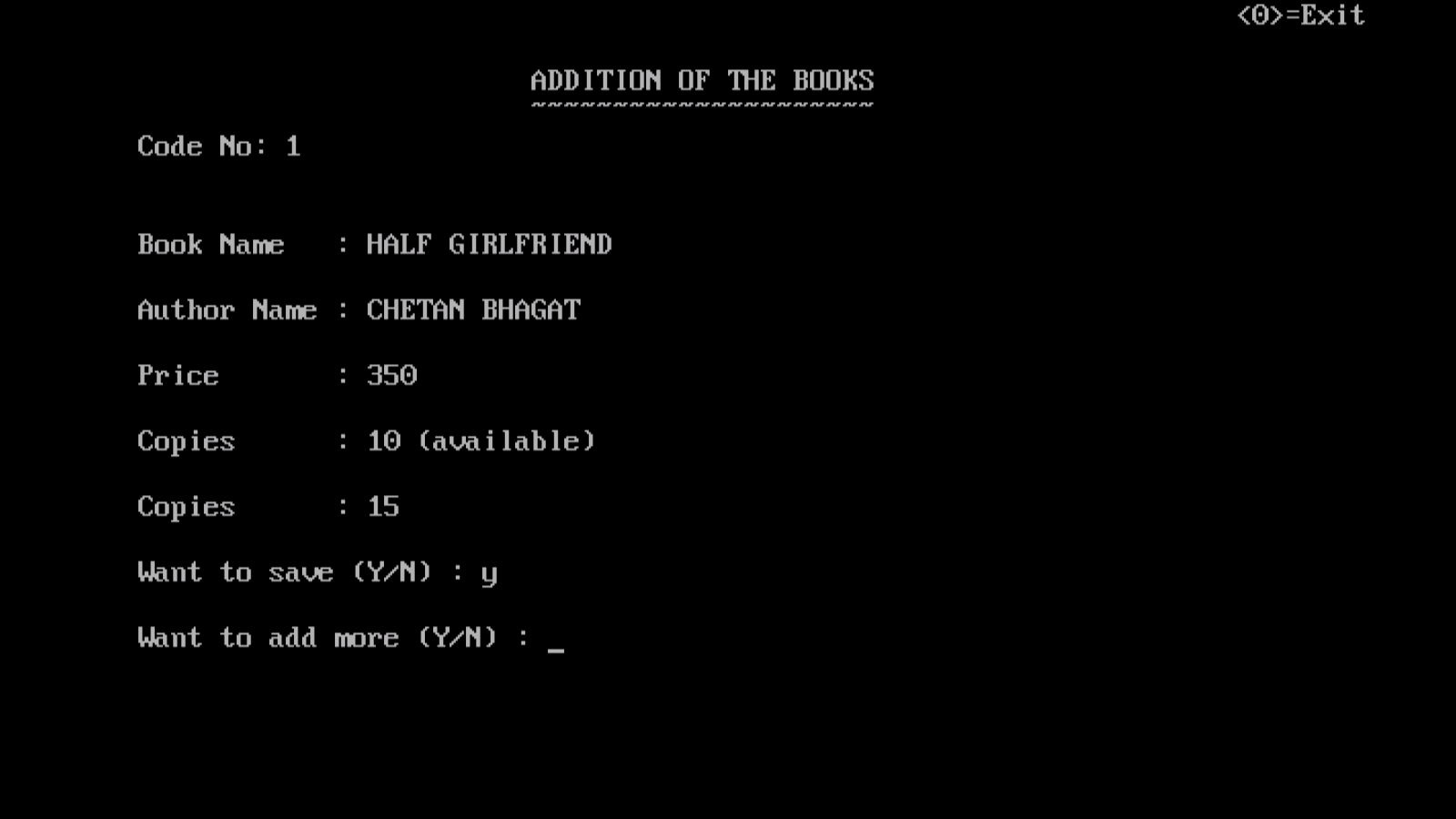
**When password “library” is not entered**

**When password is entered correctly** 

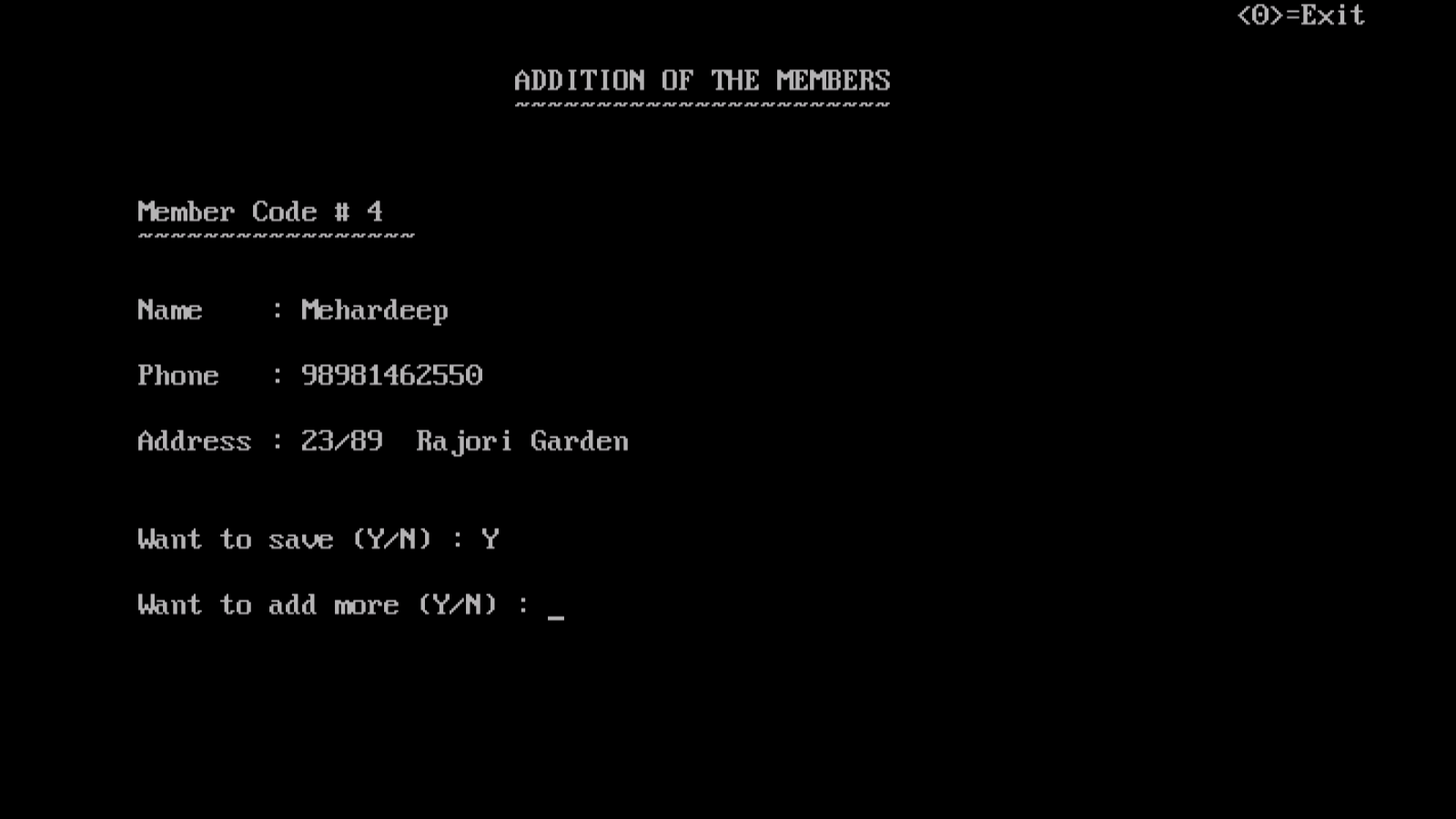
**MENU**



**OPTION:1st INTRODUCTION**

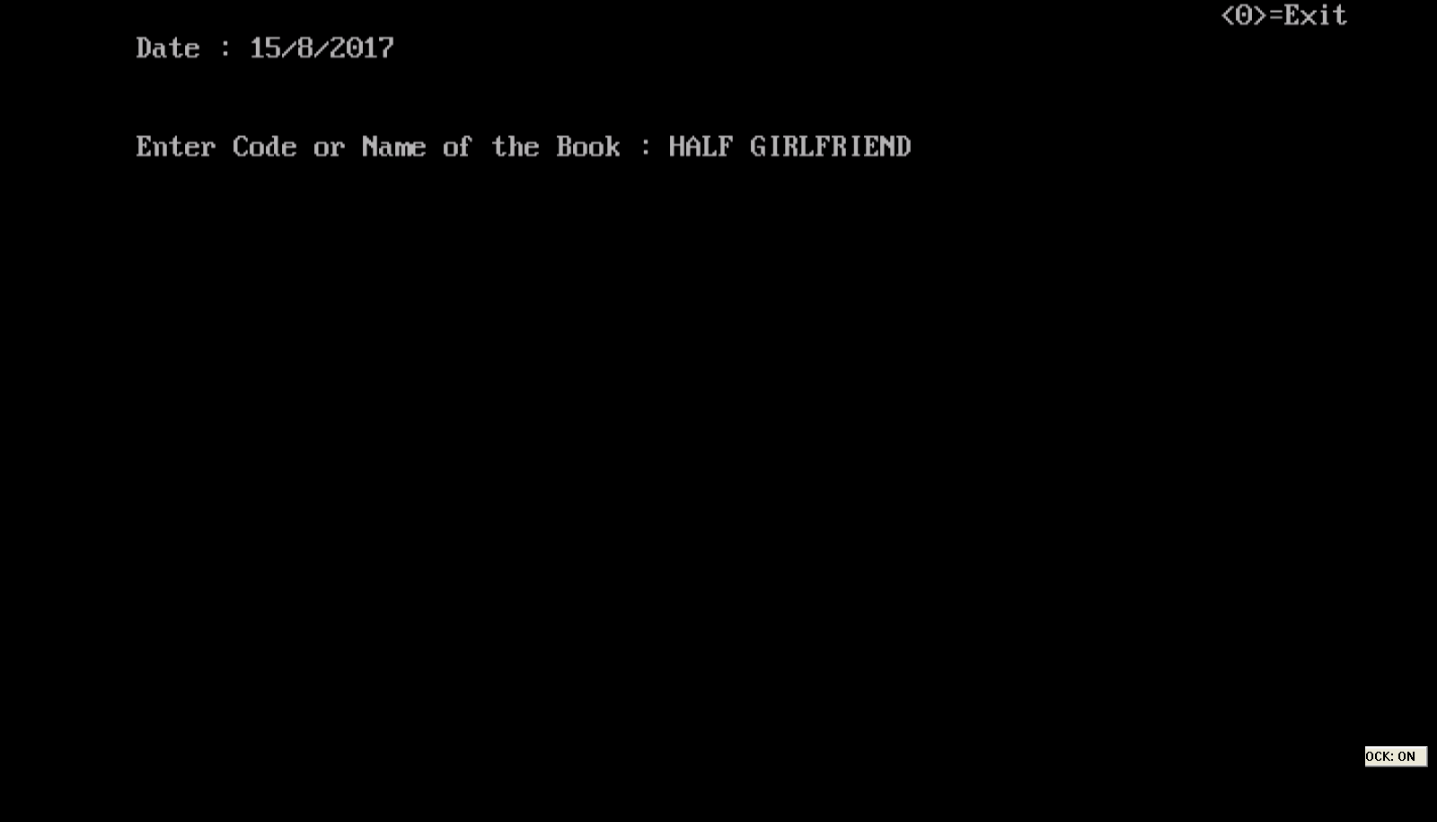
**OPTION:2ND ADD NEW BOOKS**

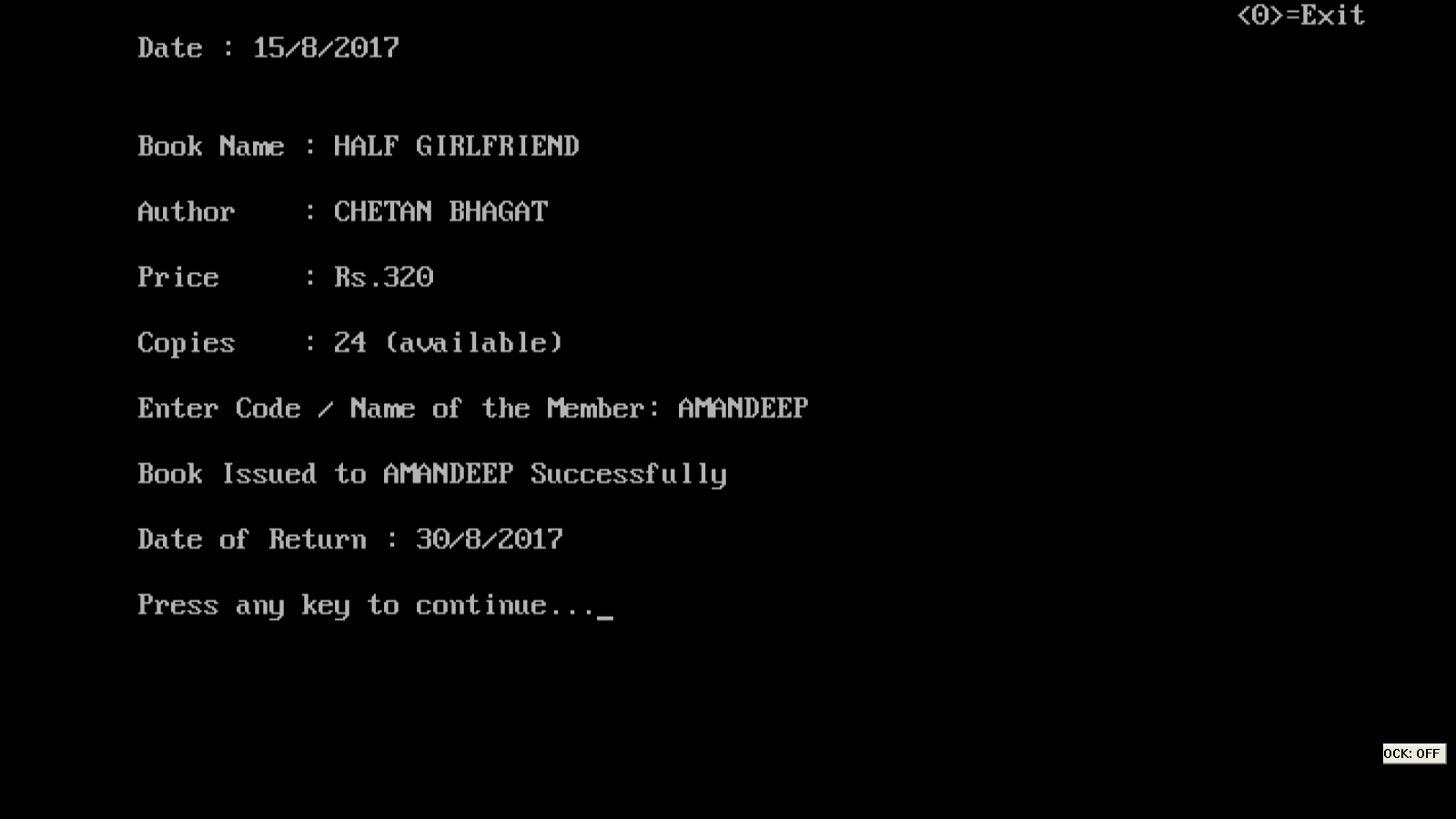
**OPTION:3RDADD NEW MEMBER**



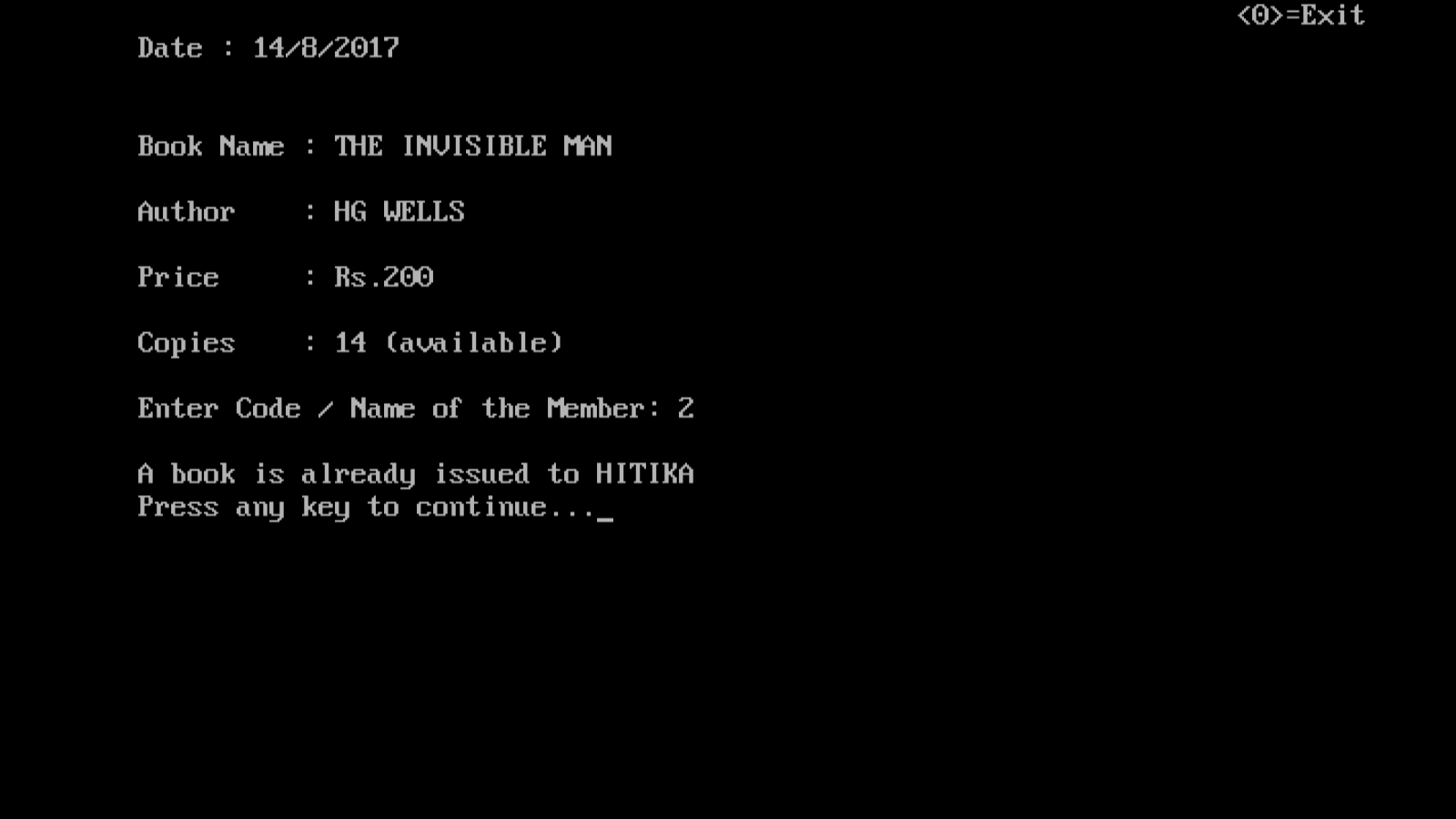
**OPTION: 4TH Issue BOOK**

**When NAME of the book to be issued is typed:-**

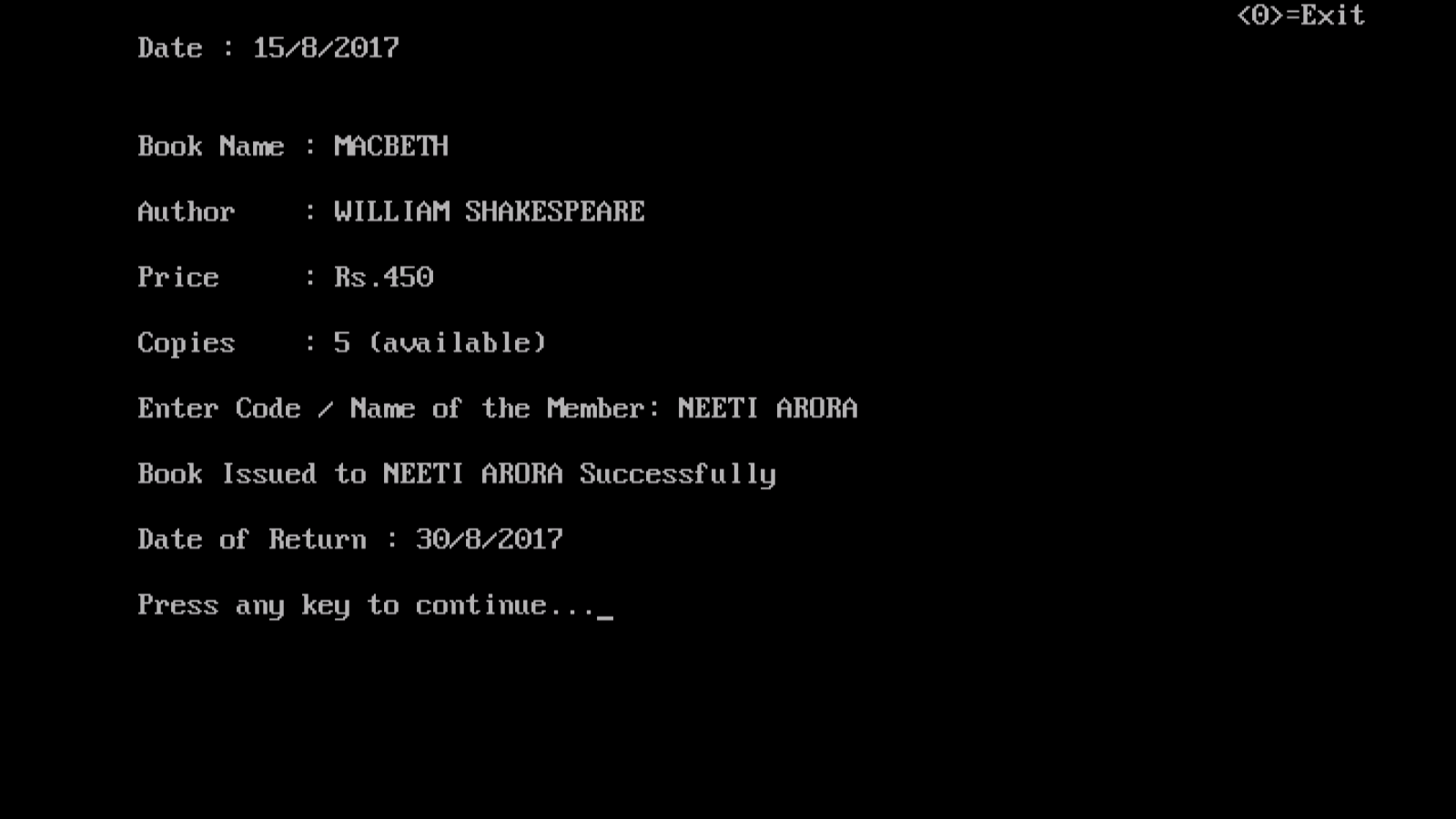
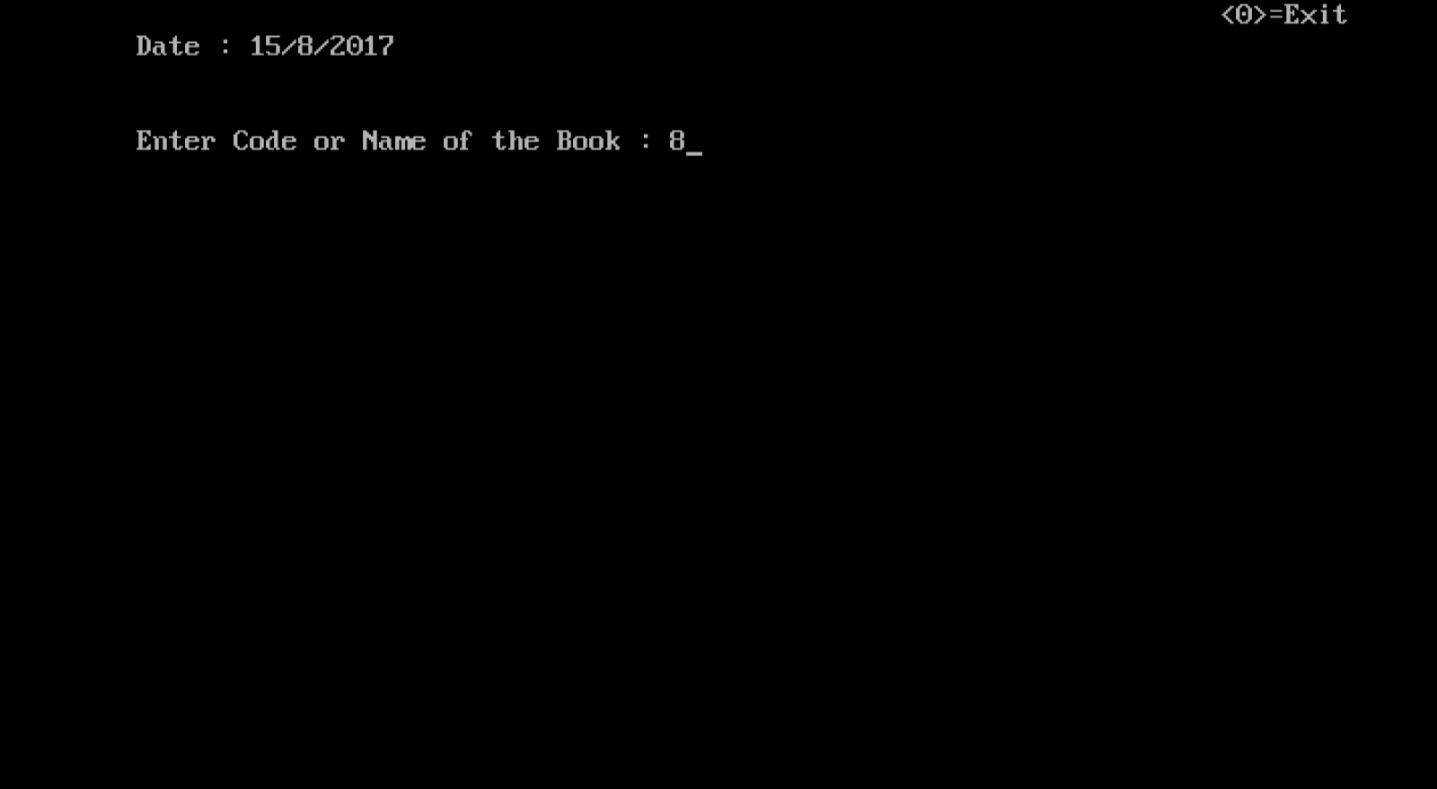




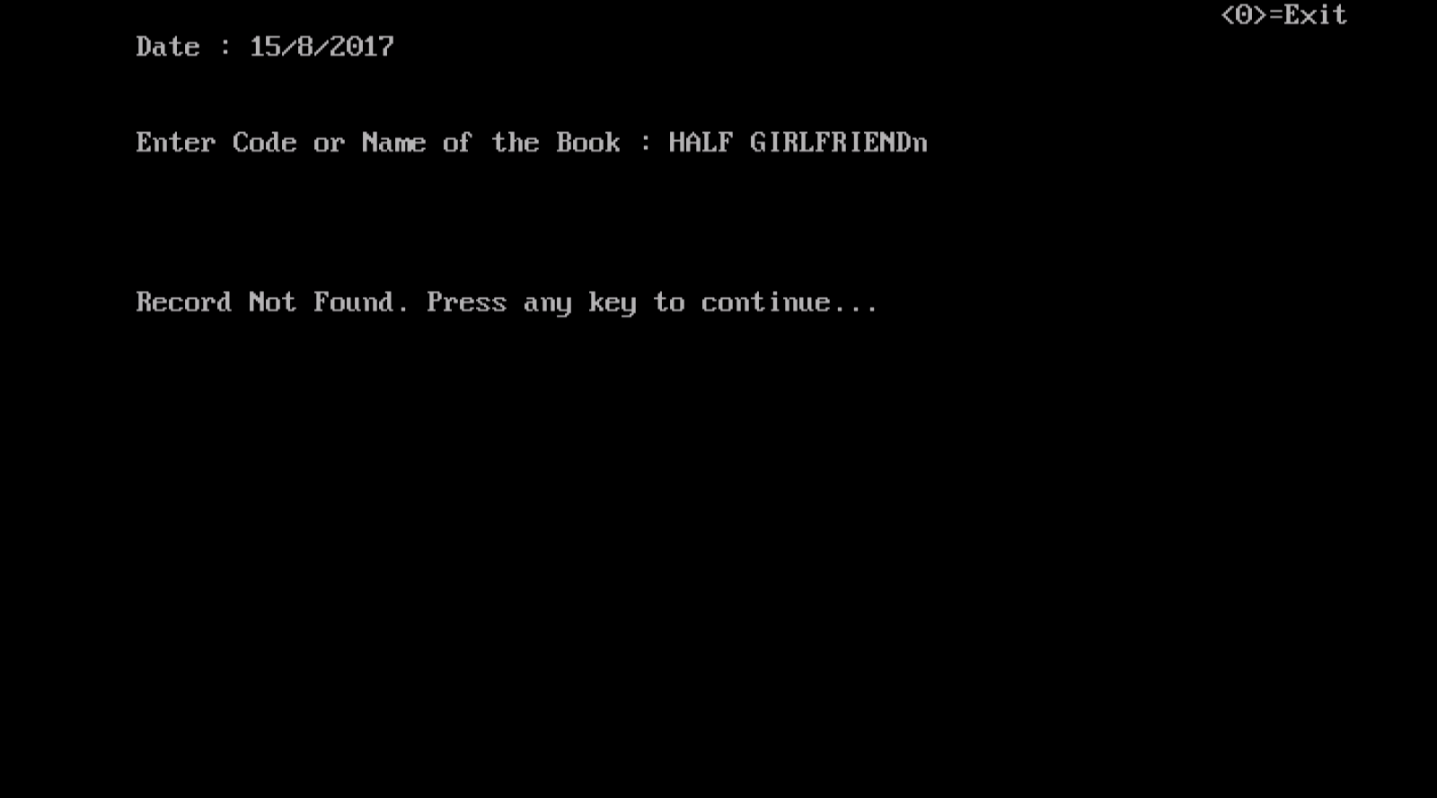
**When a member already having issued a book tries to issue another one:-**



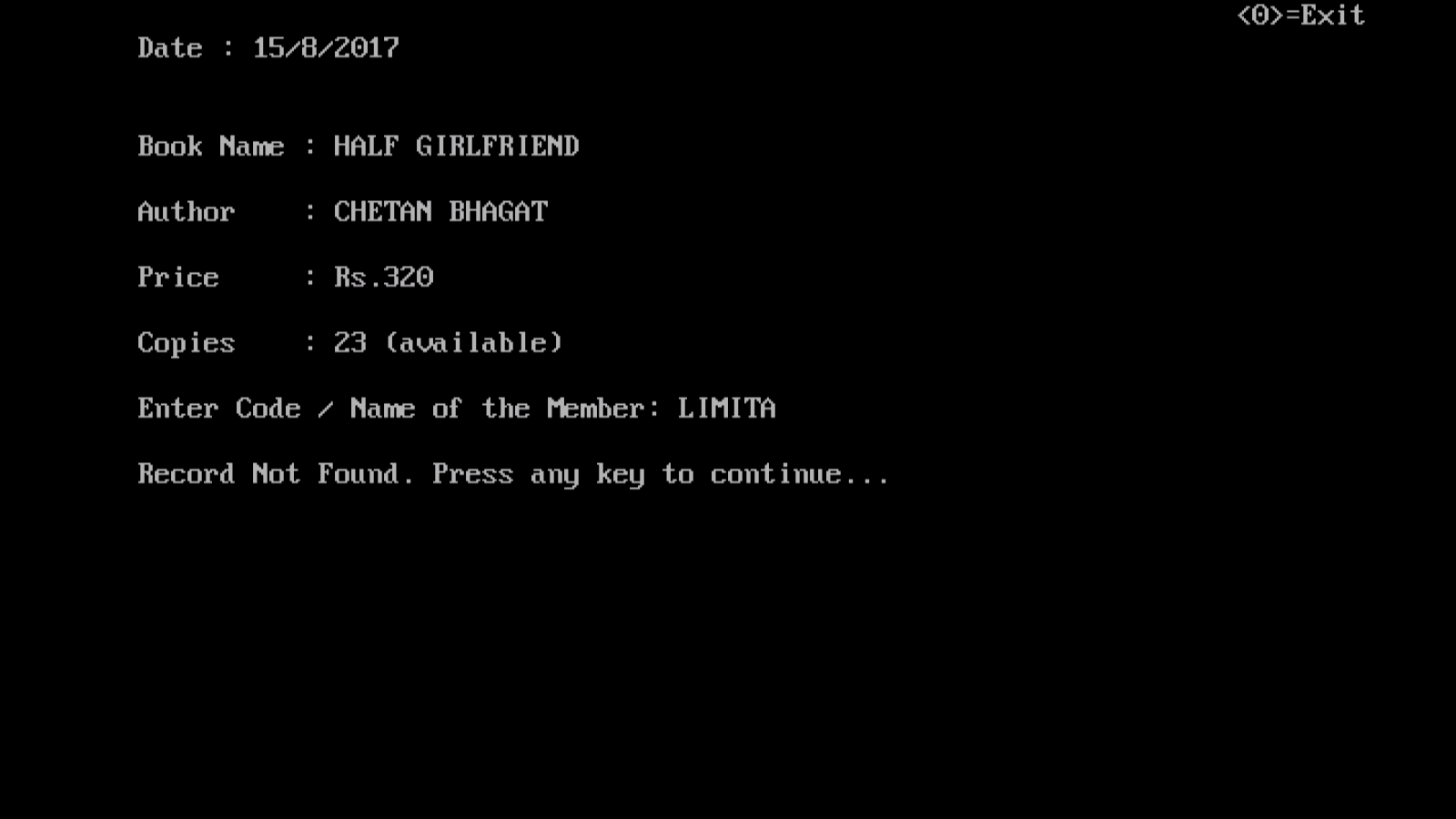
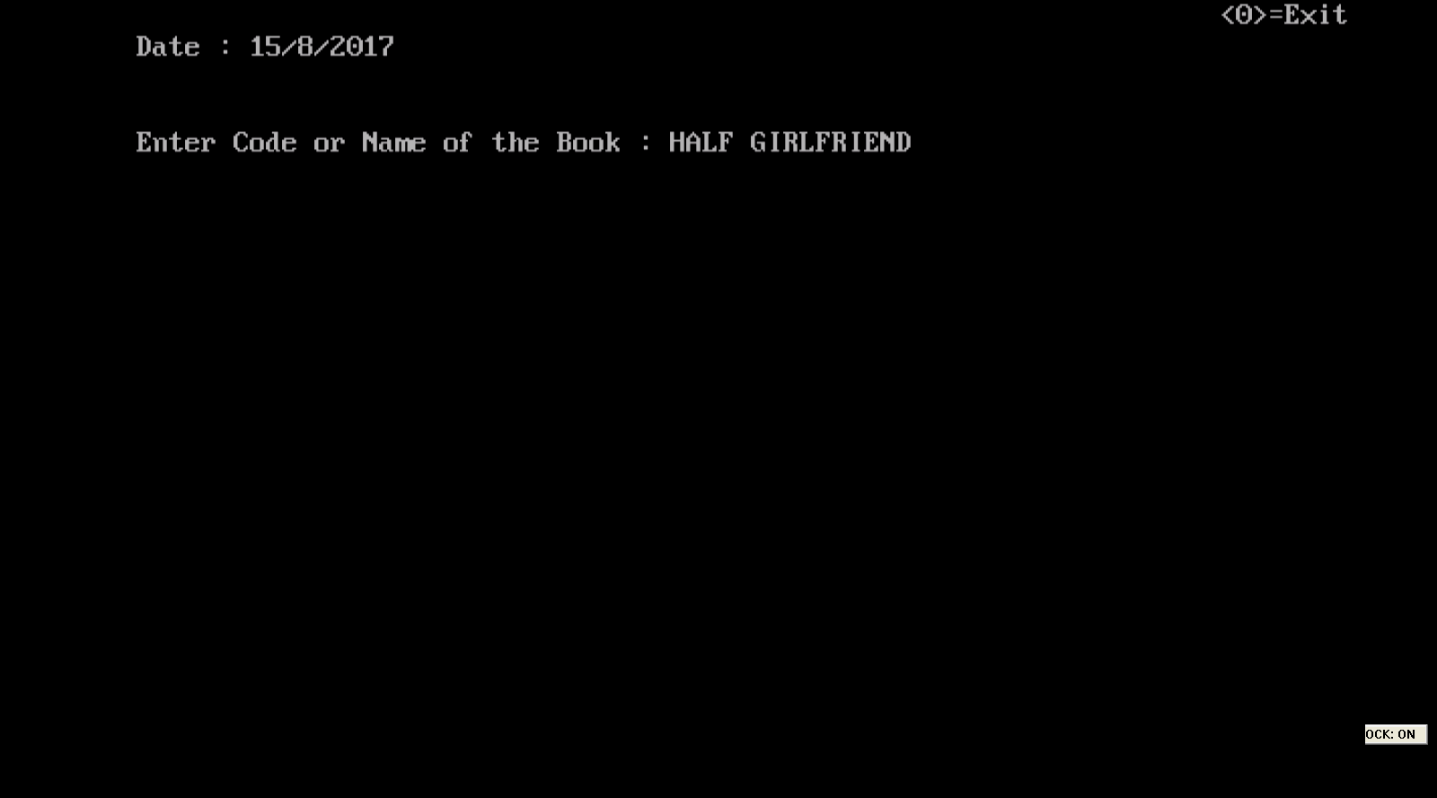
**When CODE of book to be issued & NAME of member is typed:-**



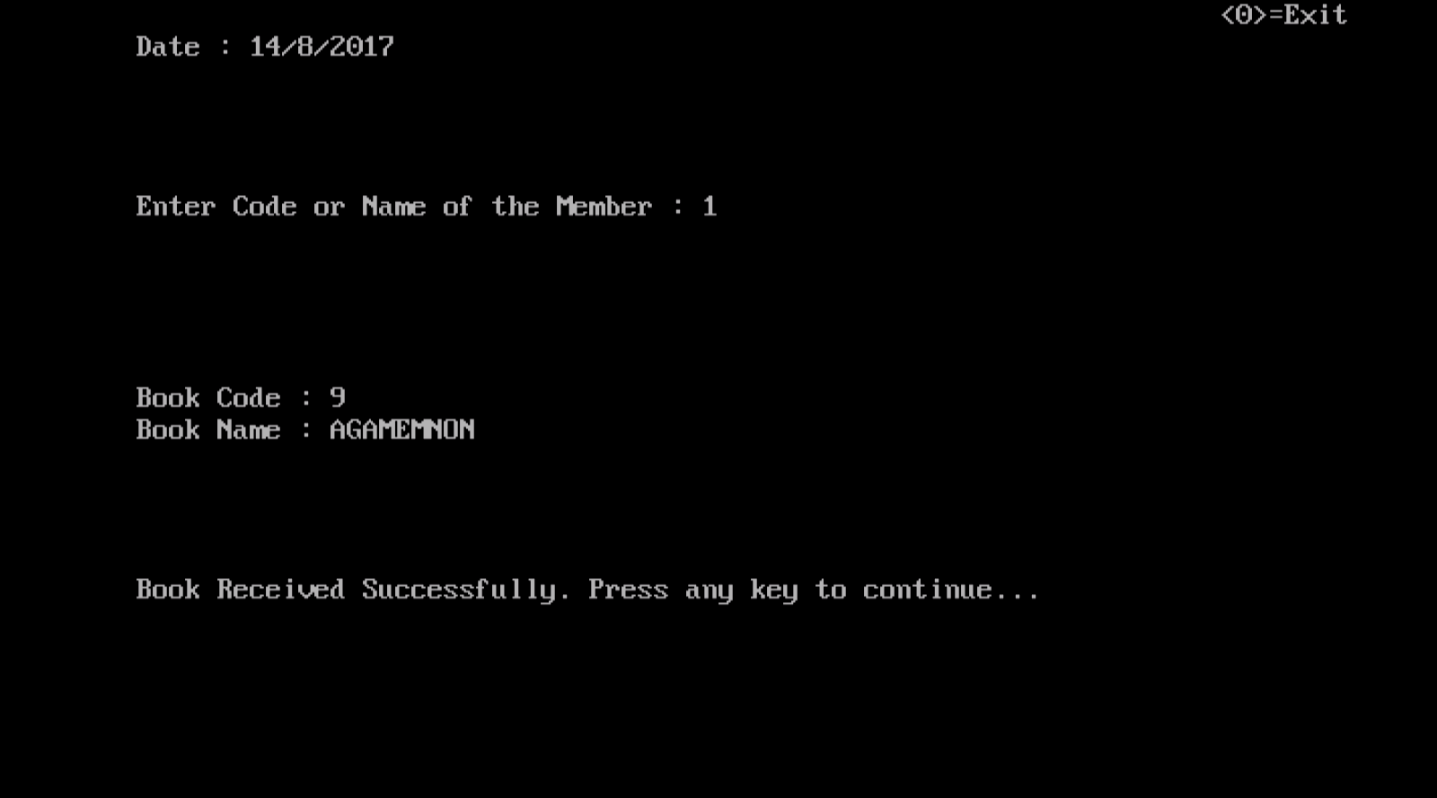
**When INVALID NAME of book to be issued is typed:-**

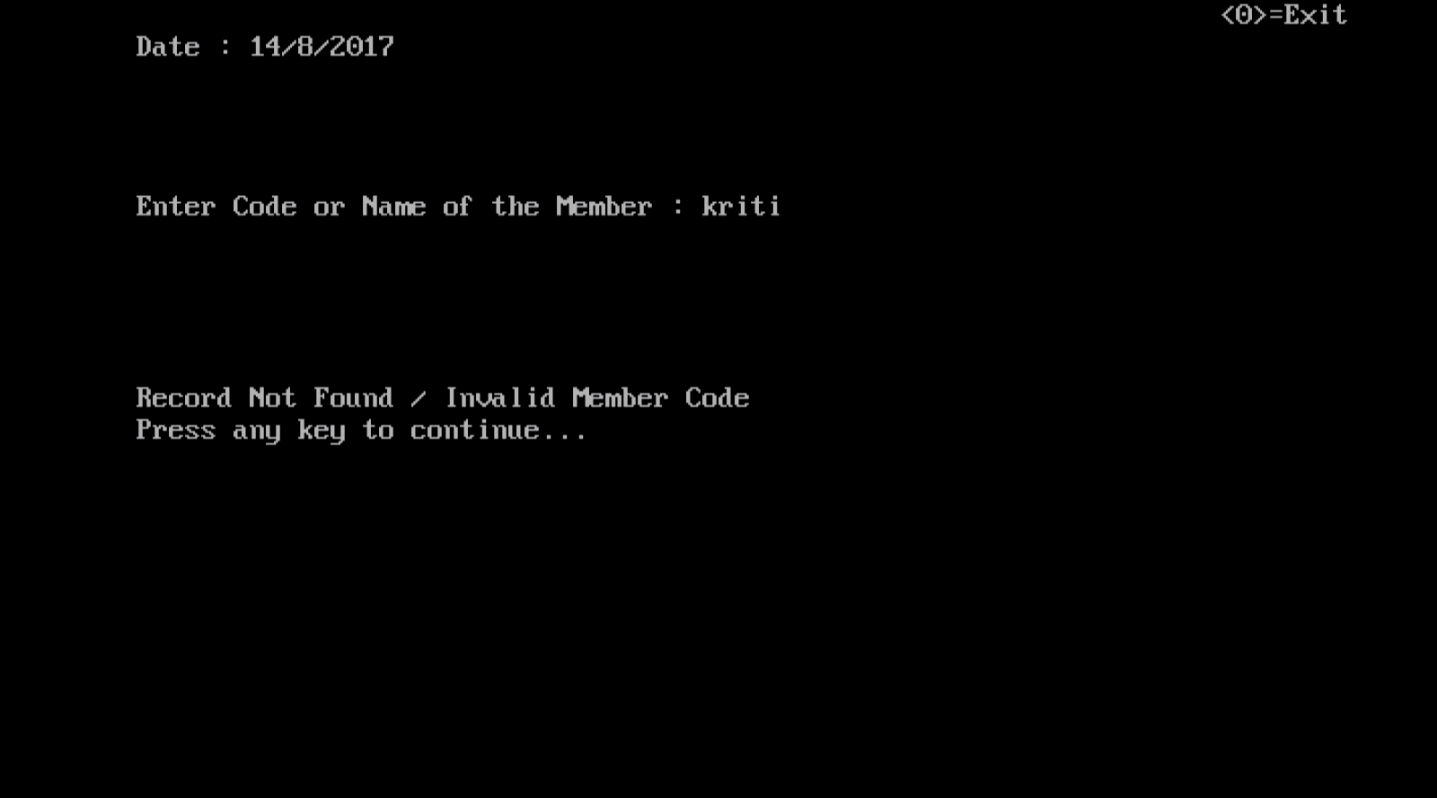


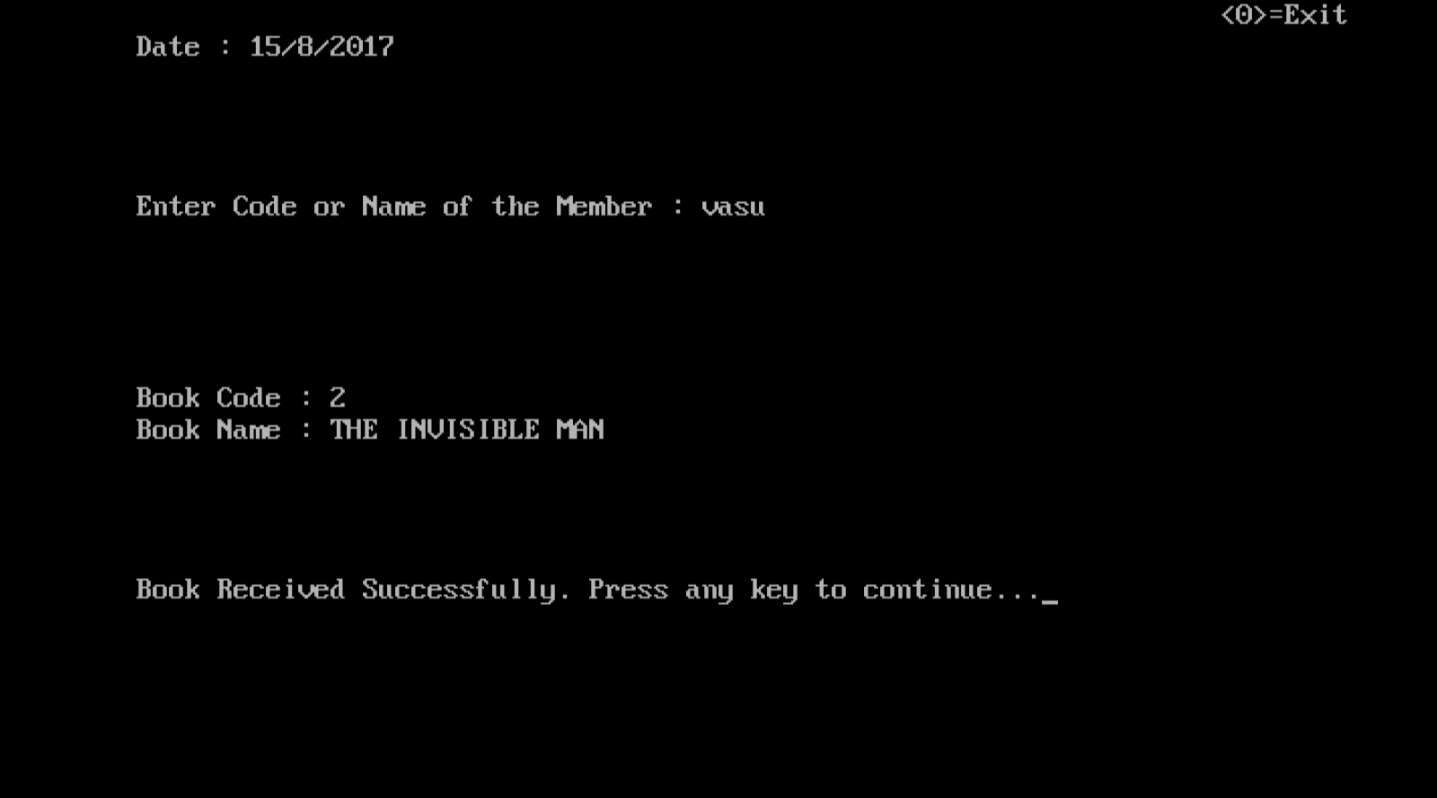
**When a non-existing member tries to issue a book:-**

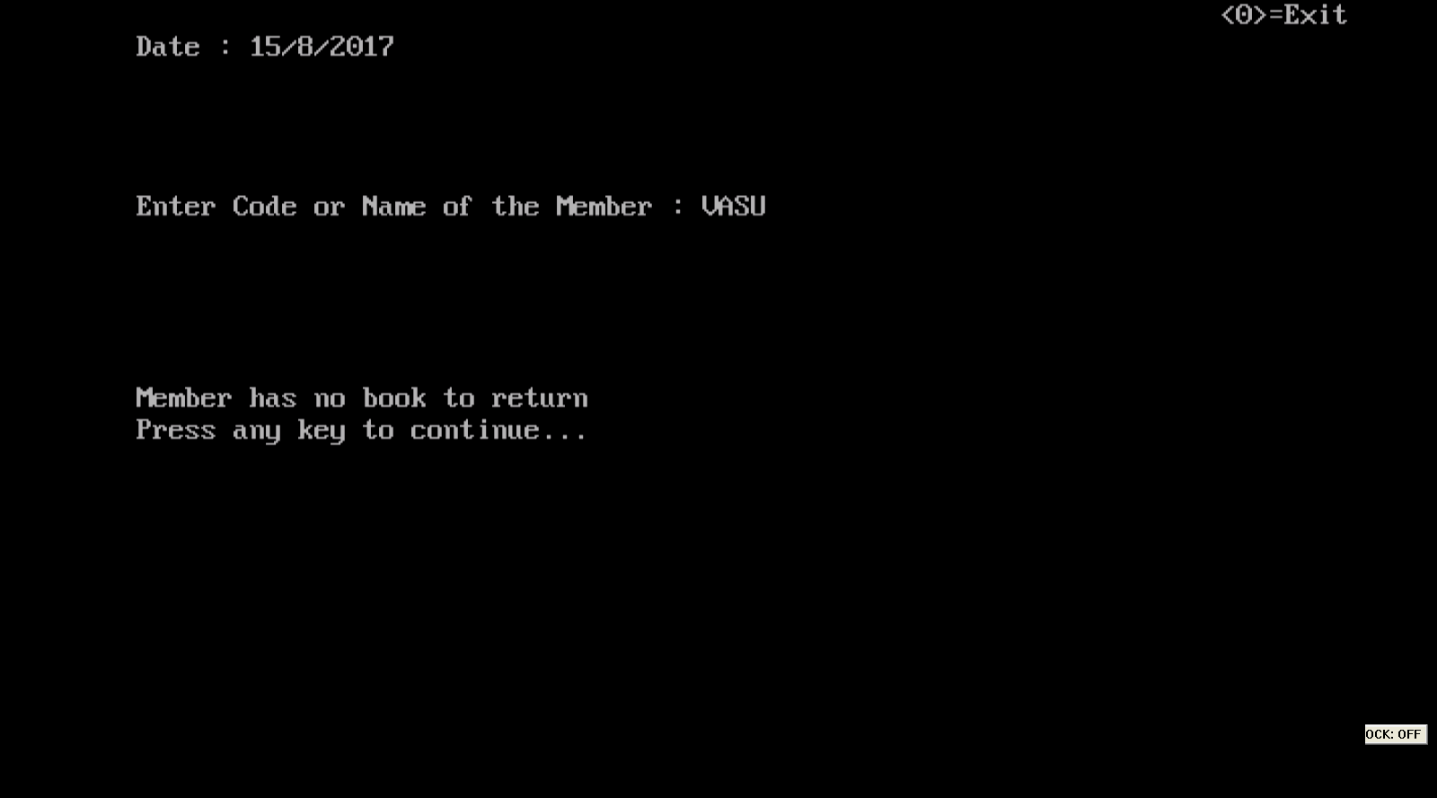


**OPTION: 5THRETURN BOOK**

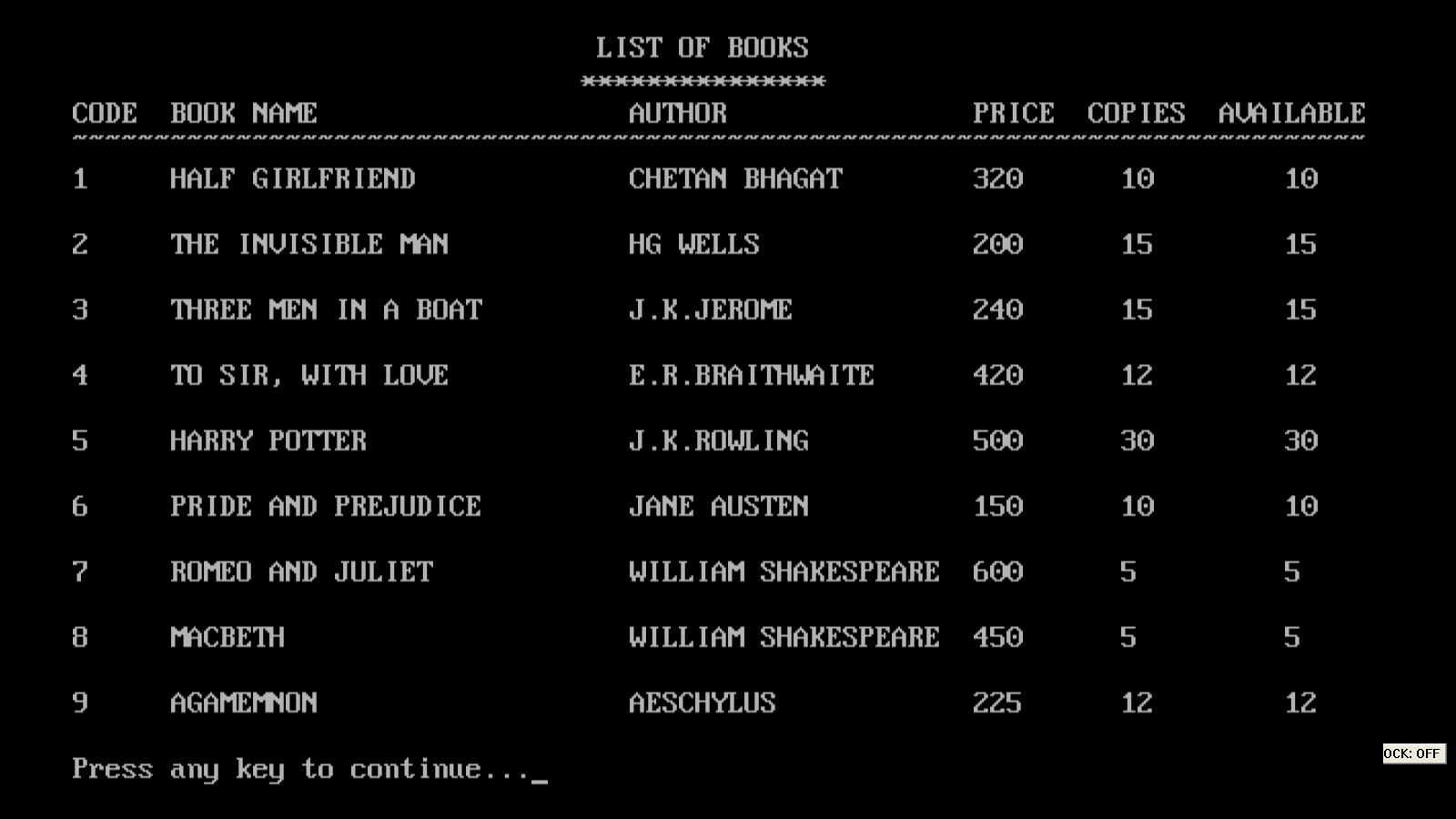
**WhenCODE of member to return book is typed:-**

**When non-existing NAME of member is typed:-**

**When NAME of member to return book is typed:-**

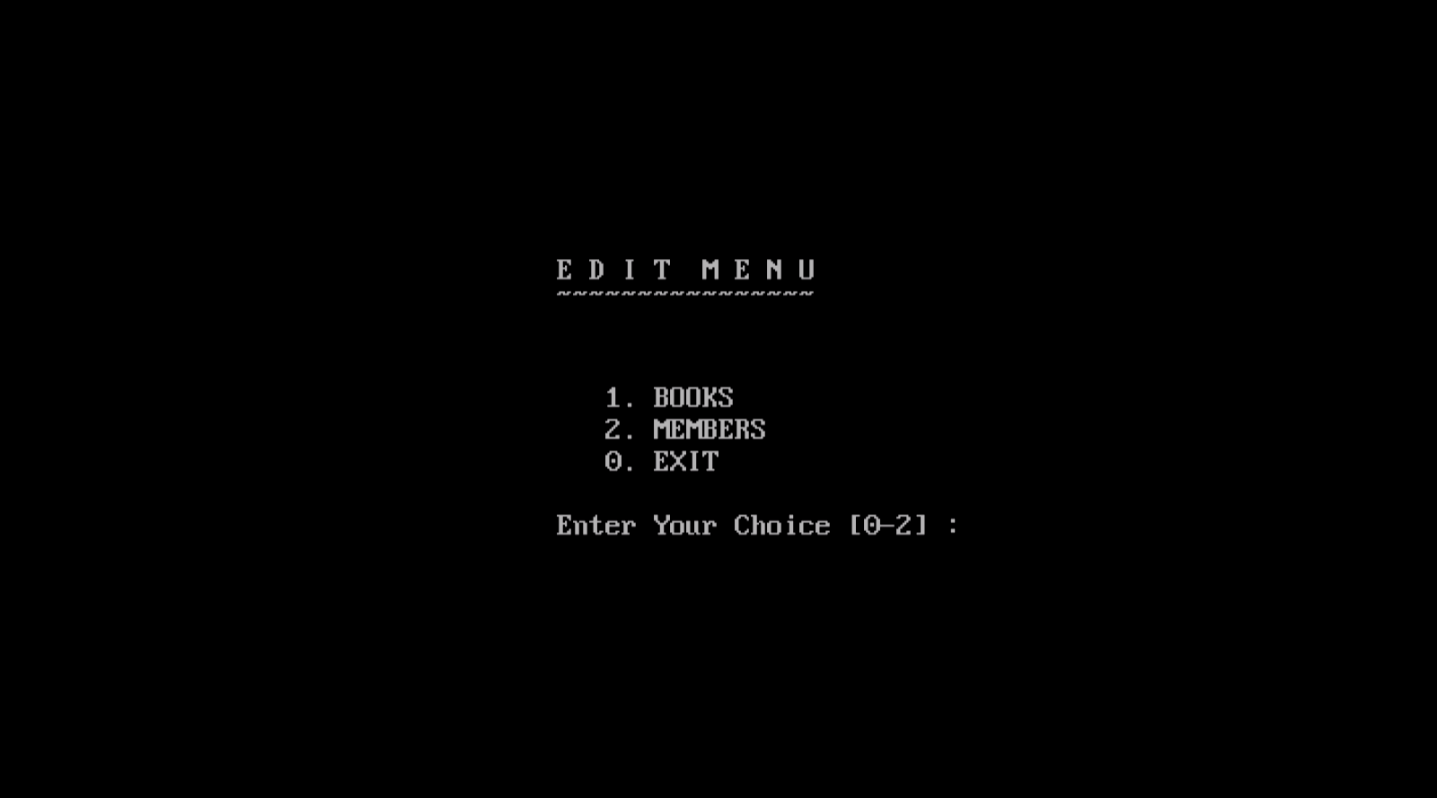
**When a member with no book issued tries to return book:-**

**OPTION:6th LIST OF BOOKS**

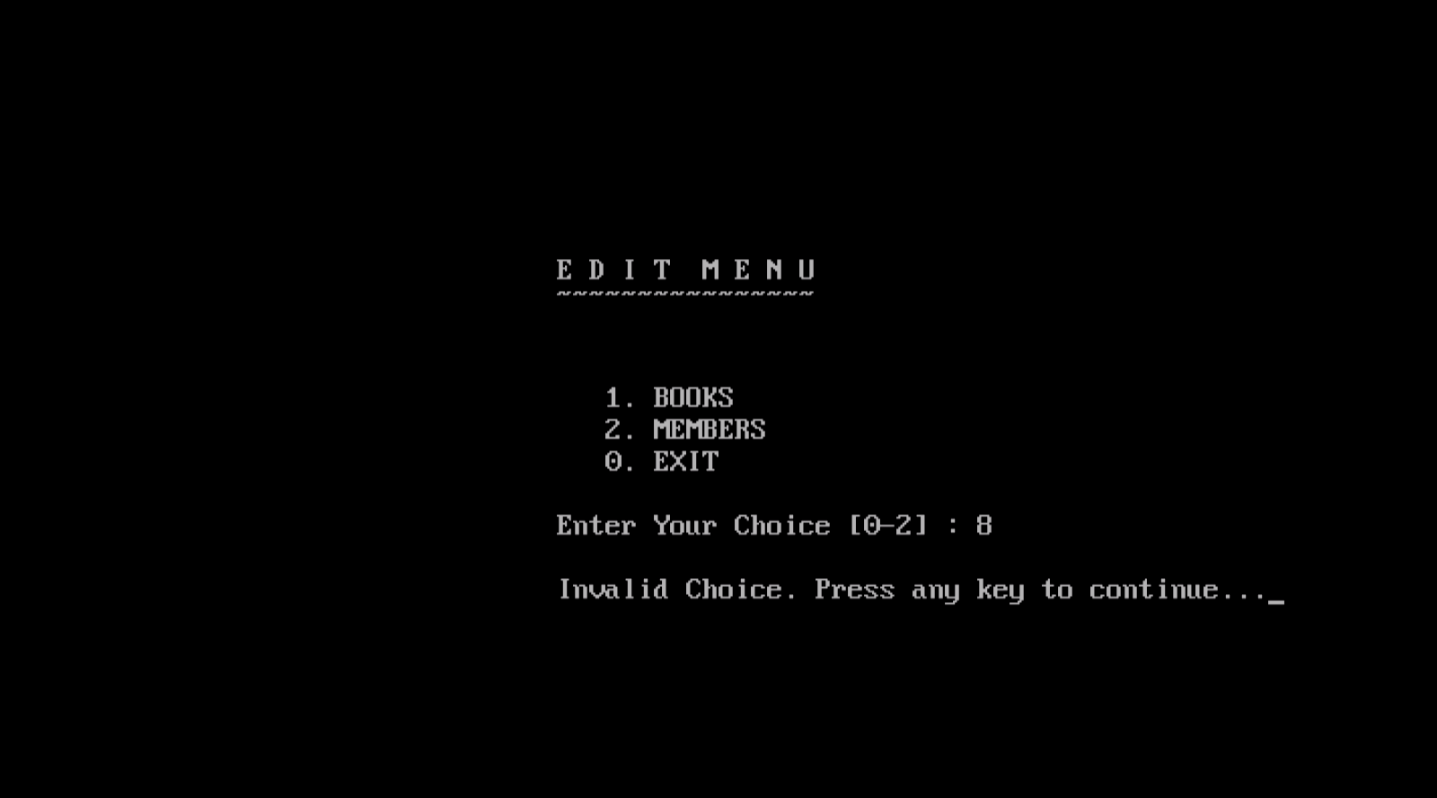


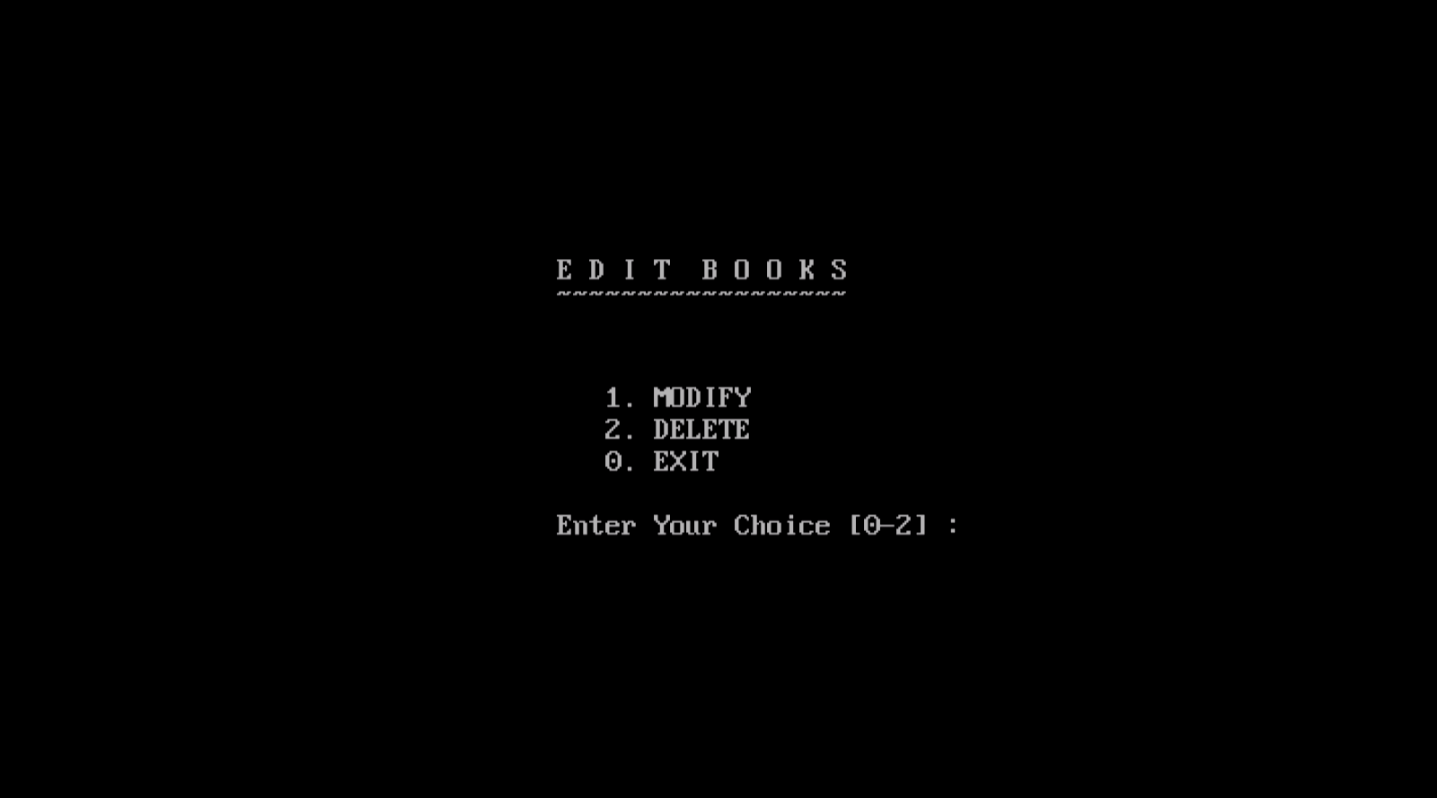
**OPTION: 7TH LIST OF MEMBERS**

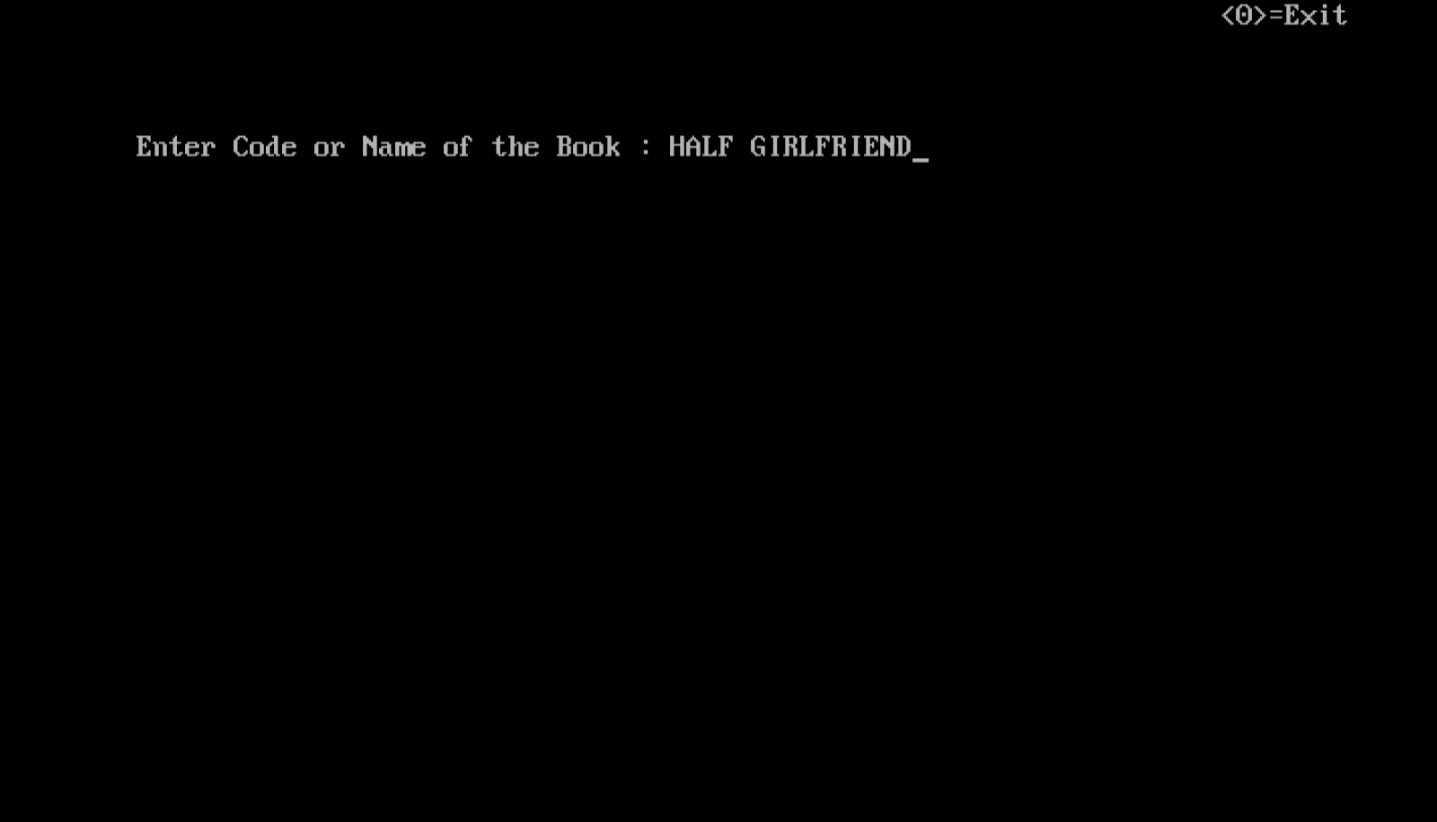


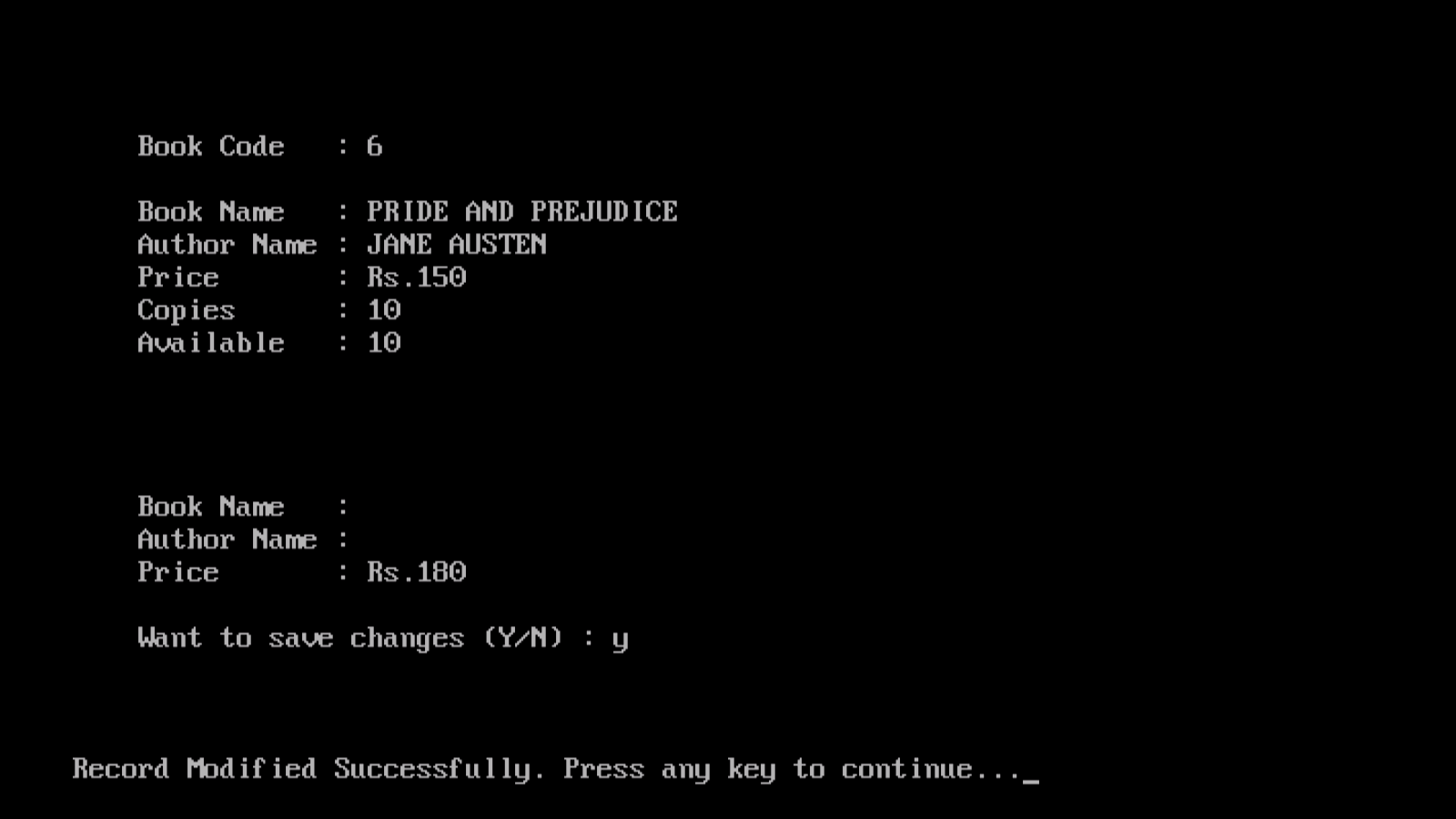
**OPTION:8thEDIT MENU**

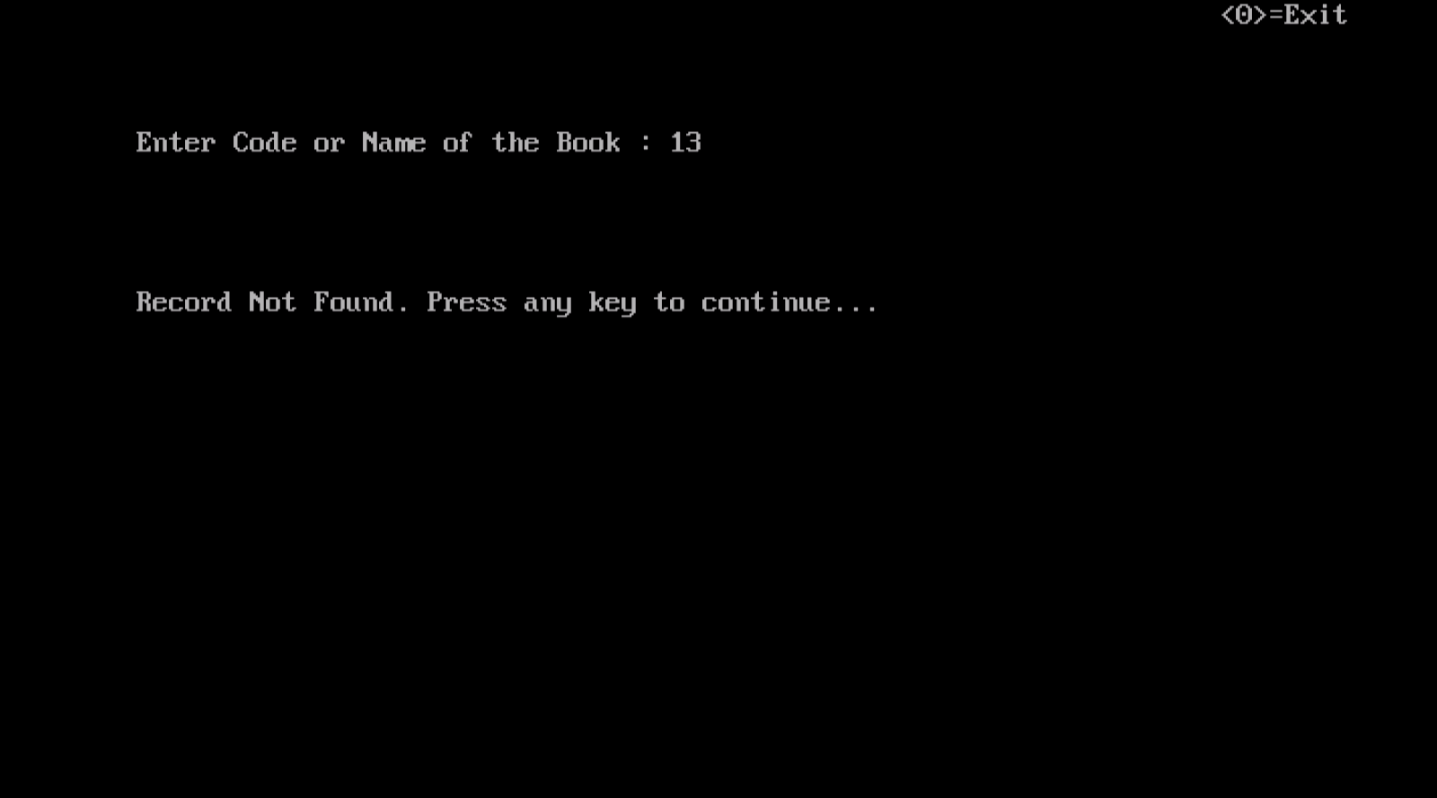
**EDIT BOOKS :**

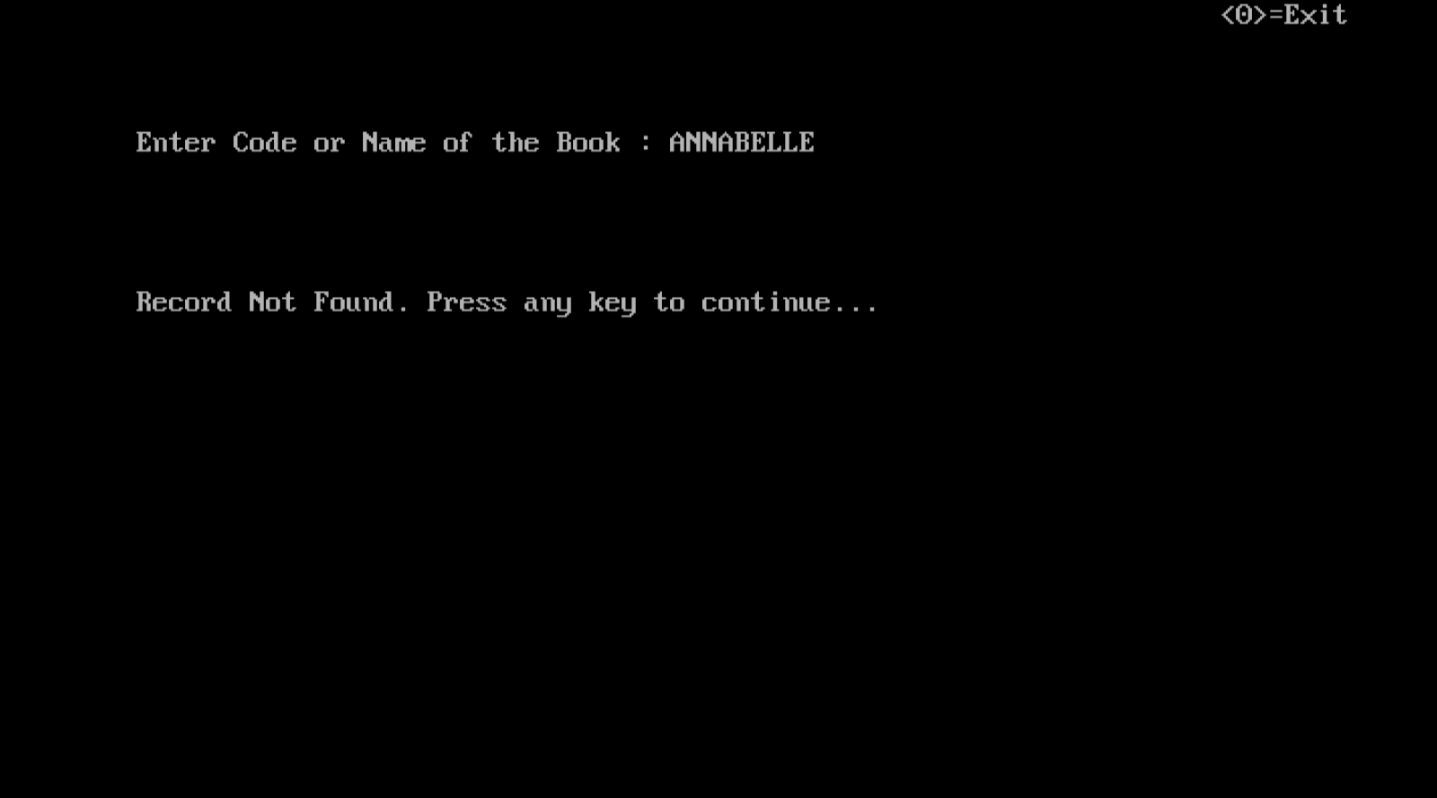
**CHOICE INCORRECT**

**CHOICE CORRECT**

**OPTION : 1st MODIFY BOOK**

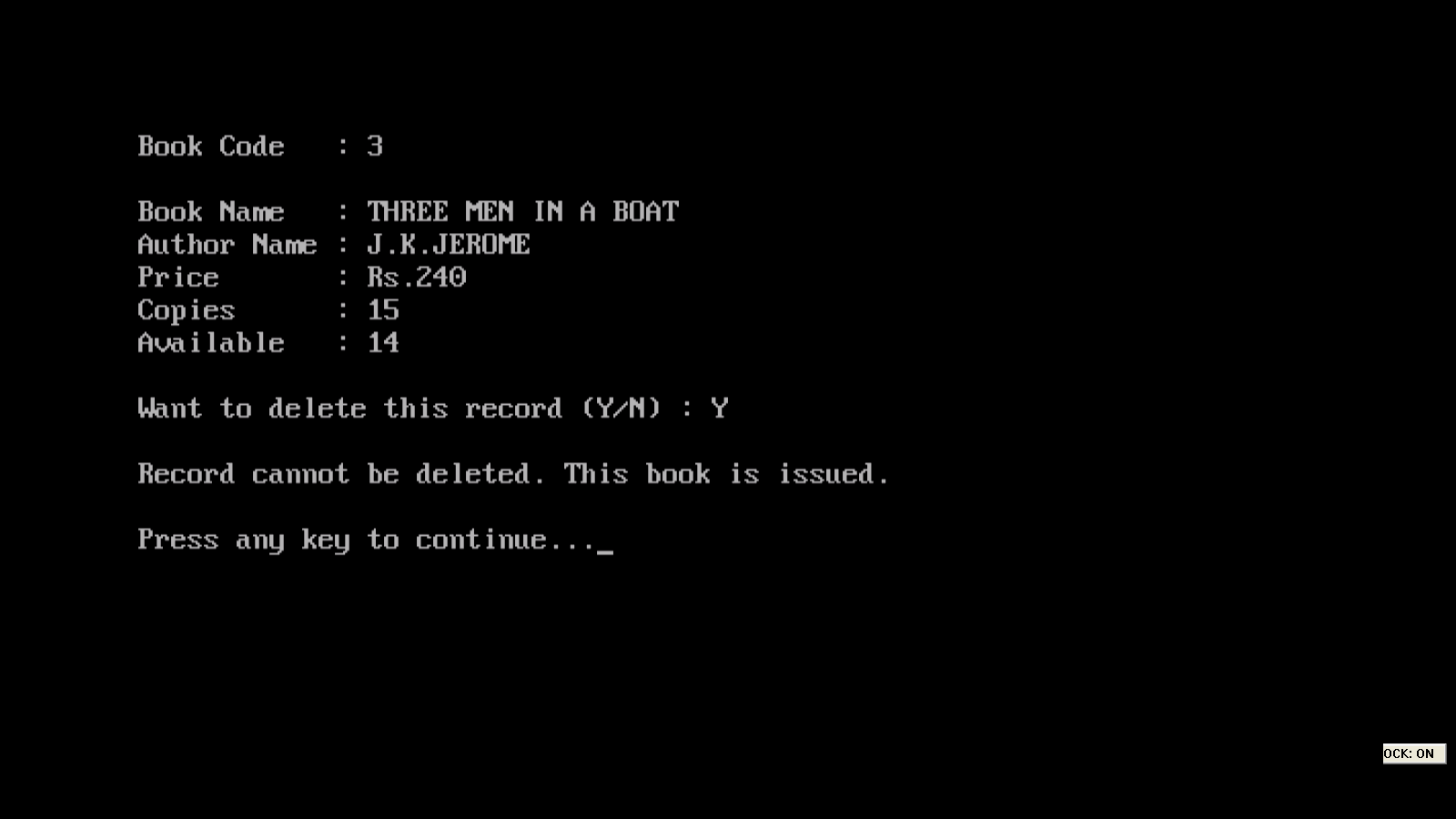
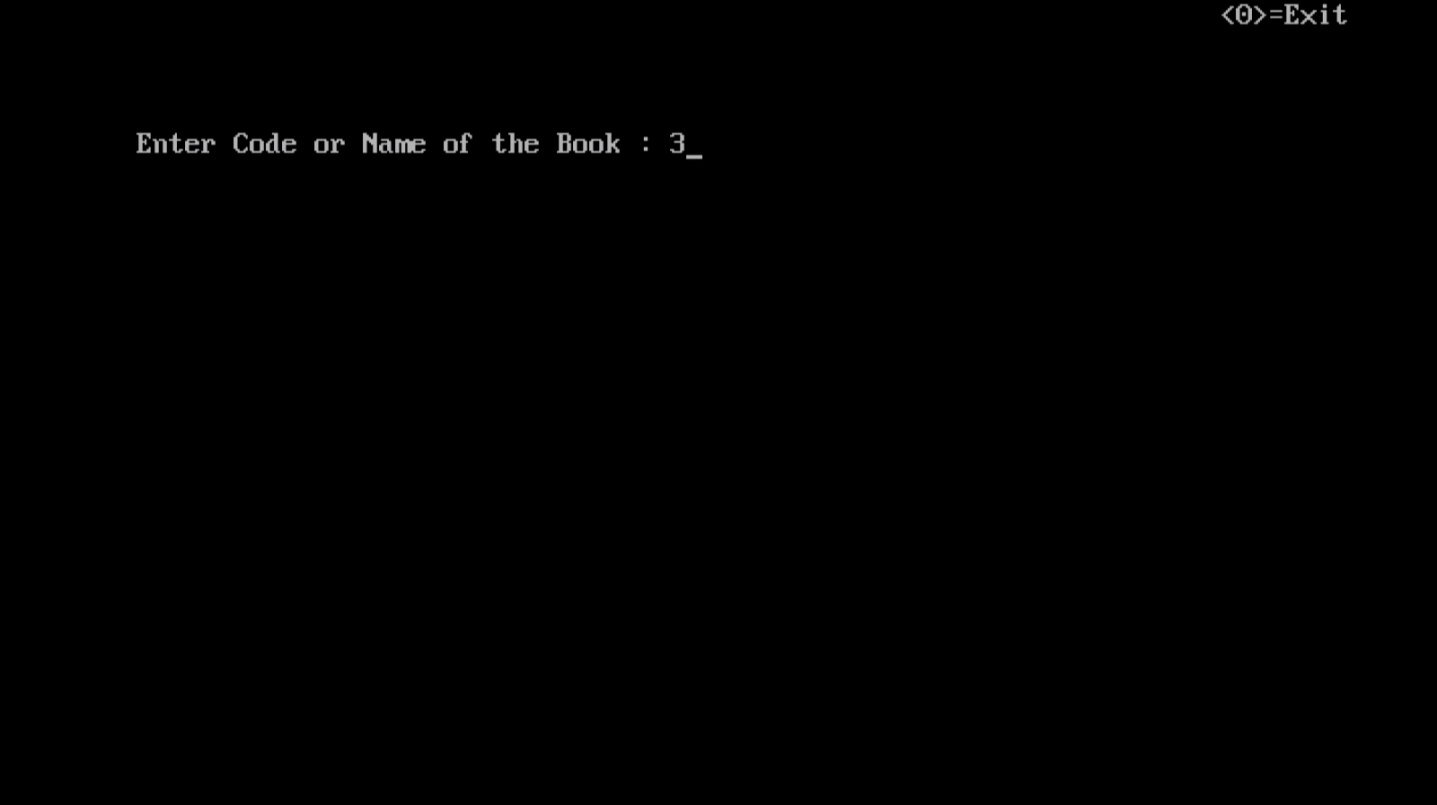
**When CODE of book whose record is to be modified is entered:-**

**When INVALID code entered:-**

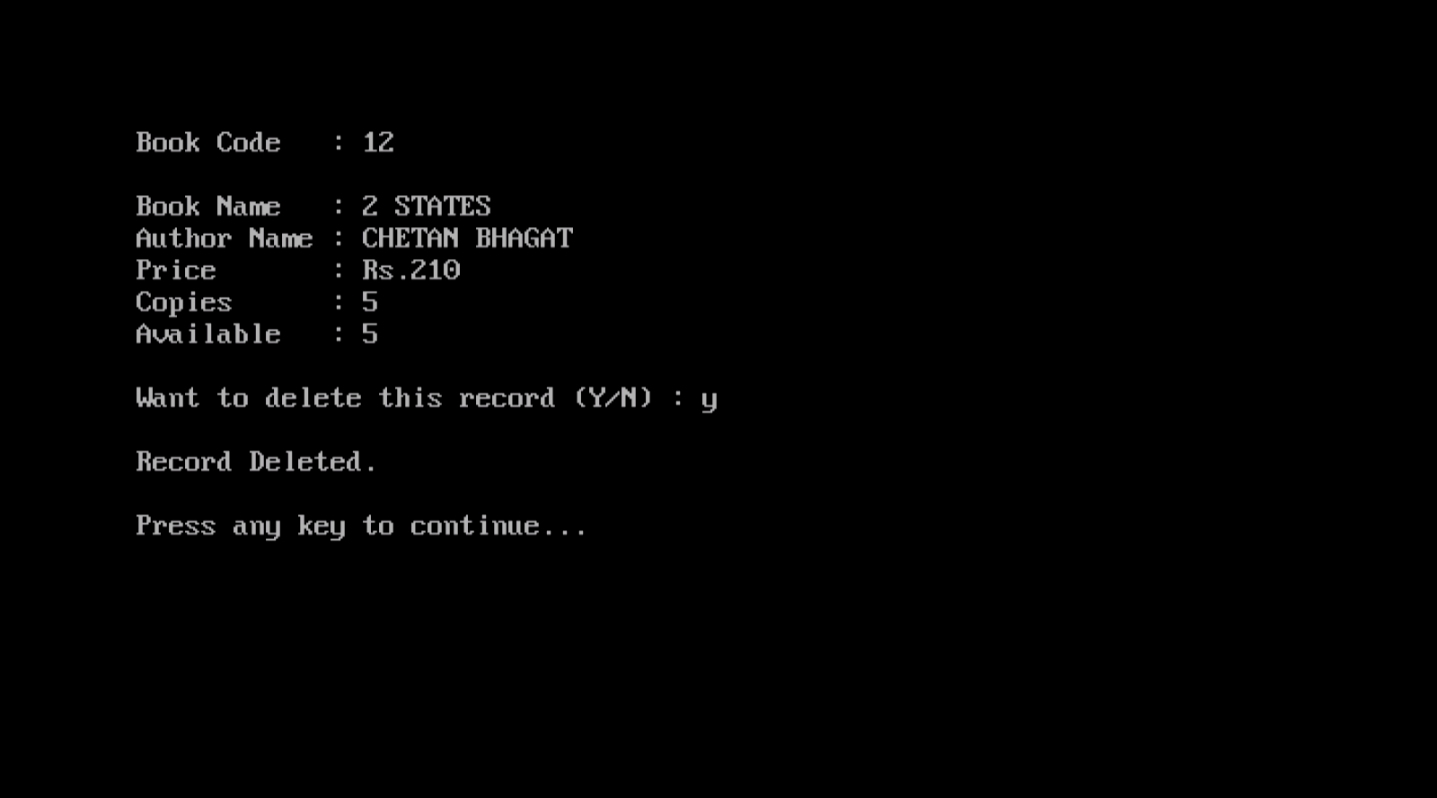
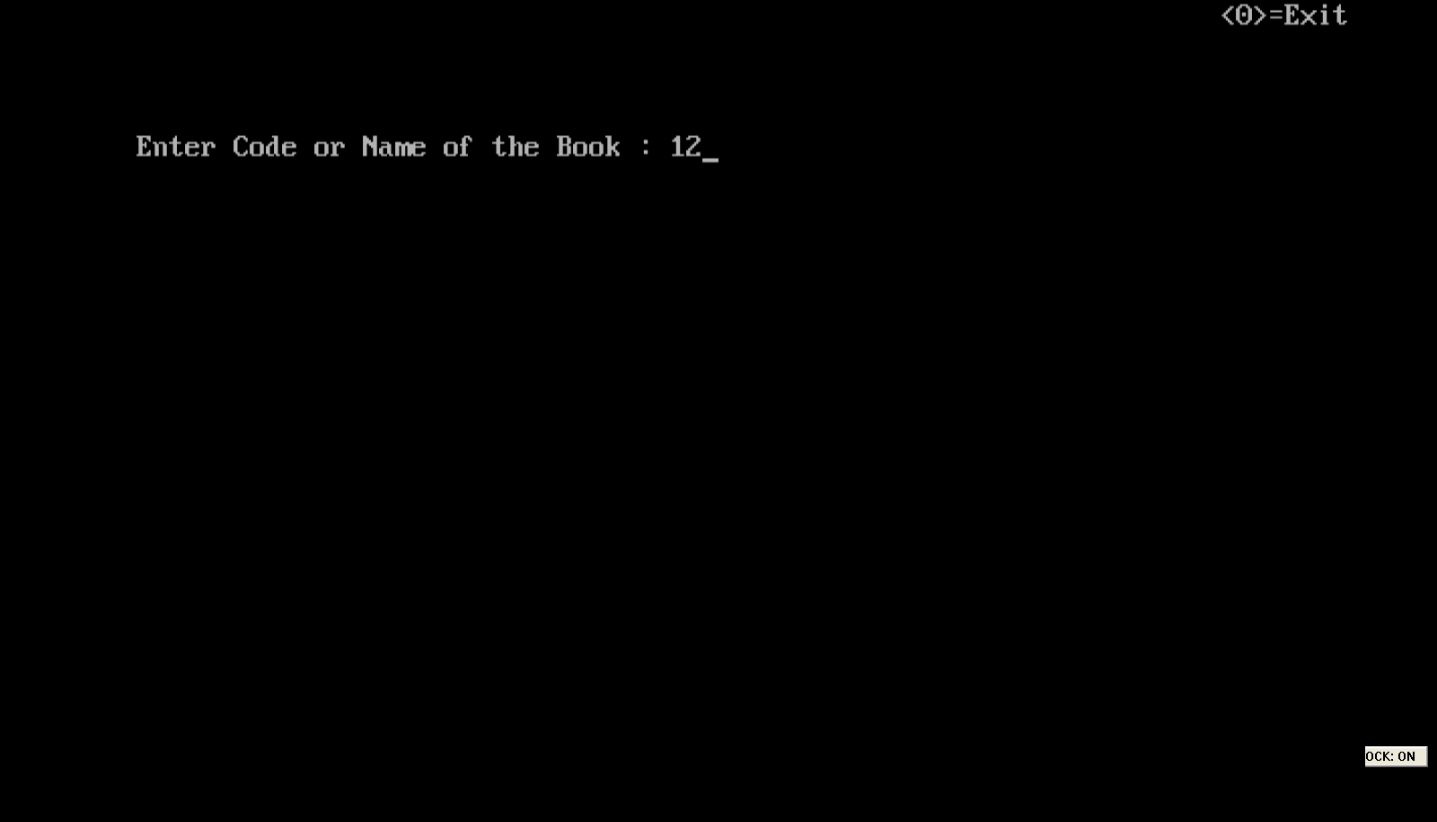
**When INVALID NAME of book is entered:-**

**OPTION:2ndDELETE BOOK**

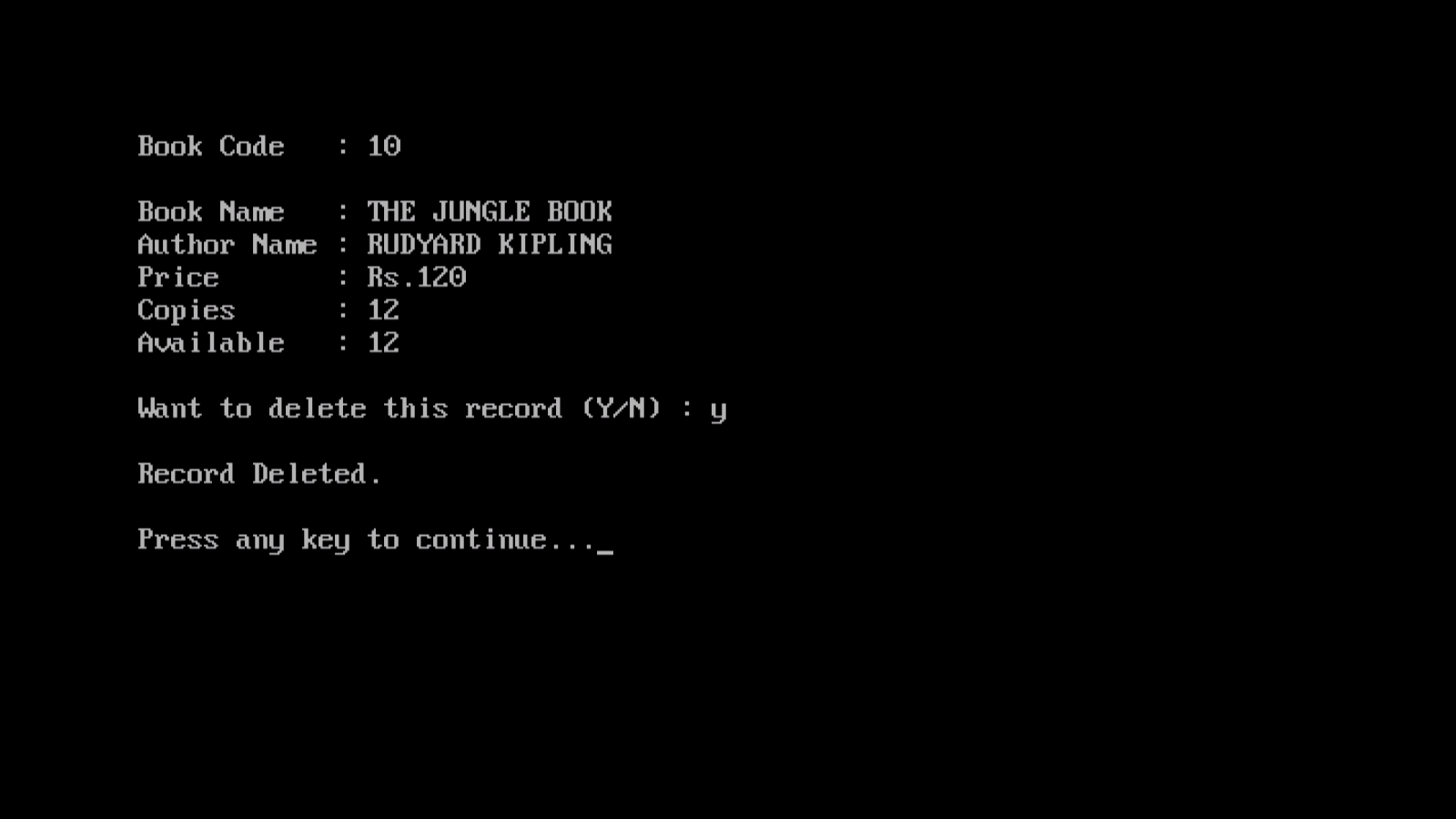
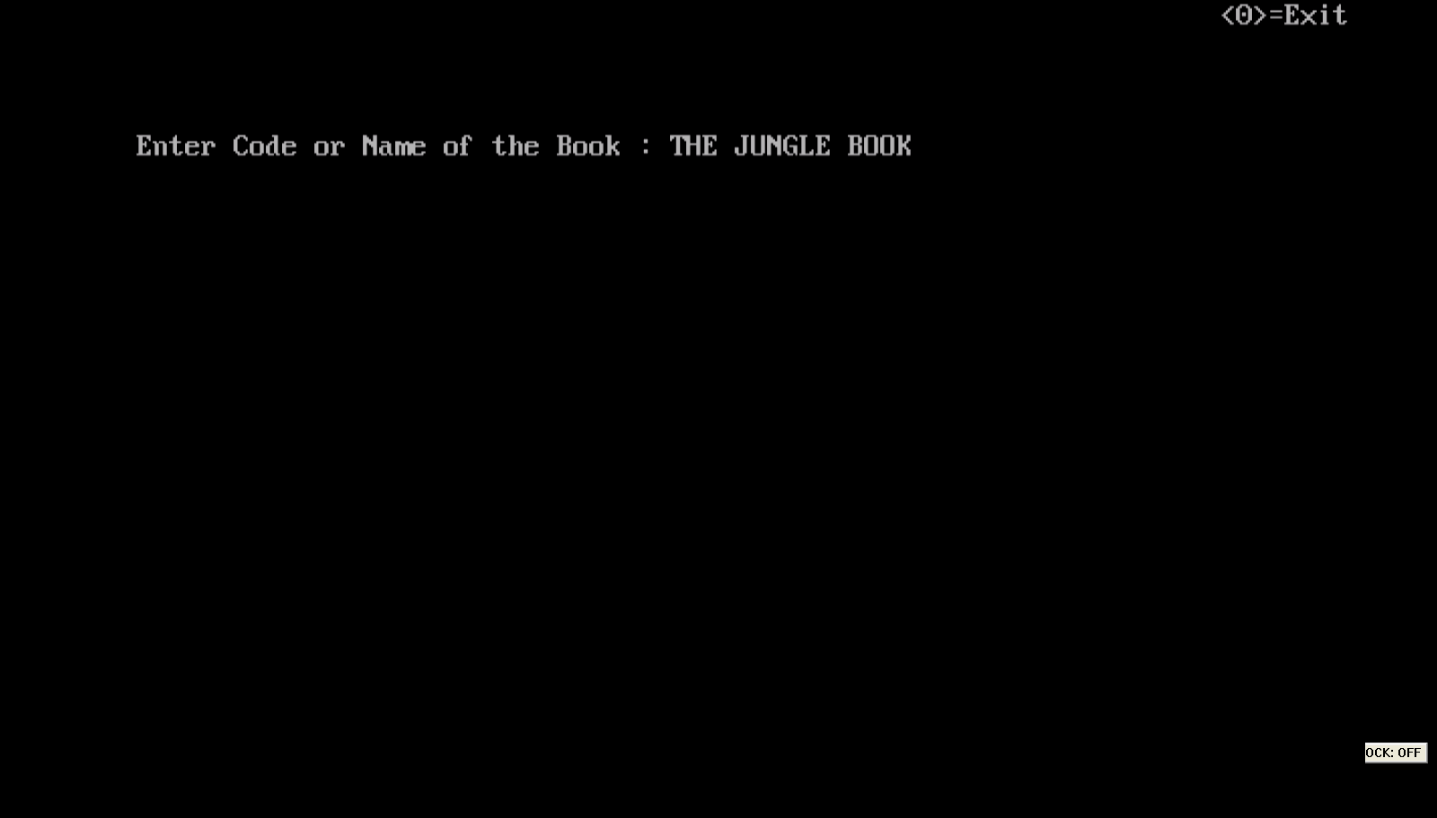
**When CODE of book to be deleted is typed and its already issued:-**

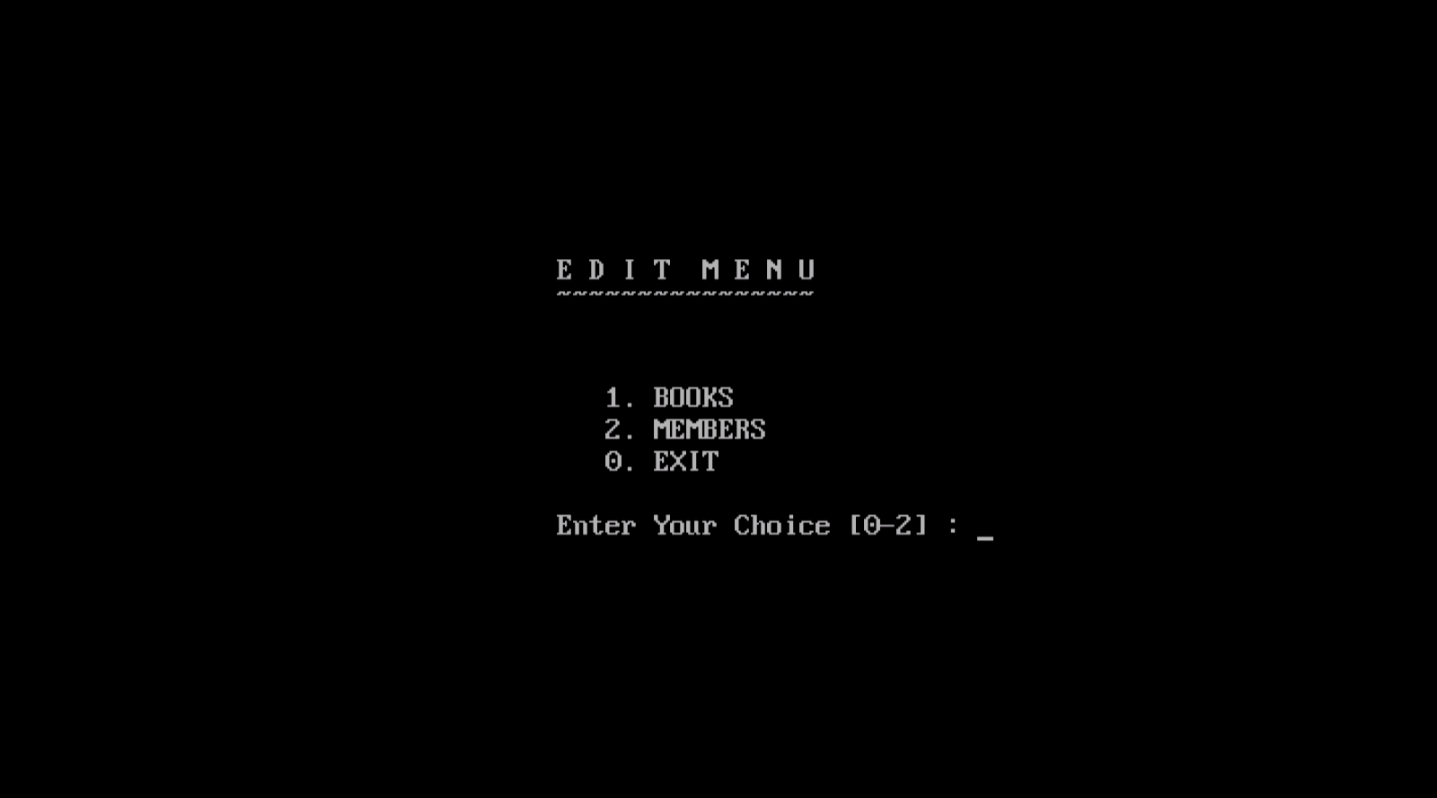


**When CODE of book to be deleted is typed:-**

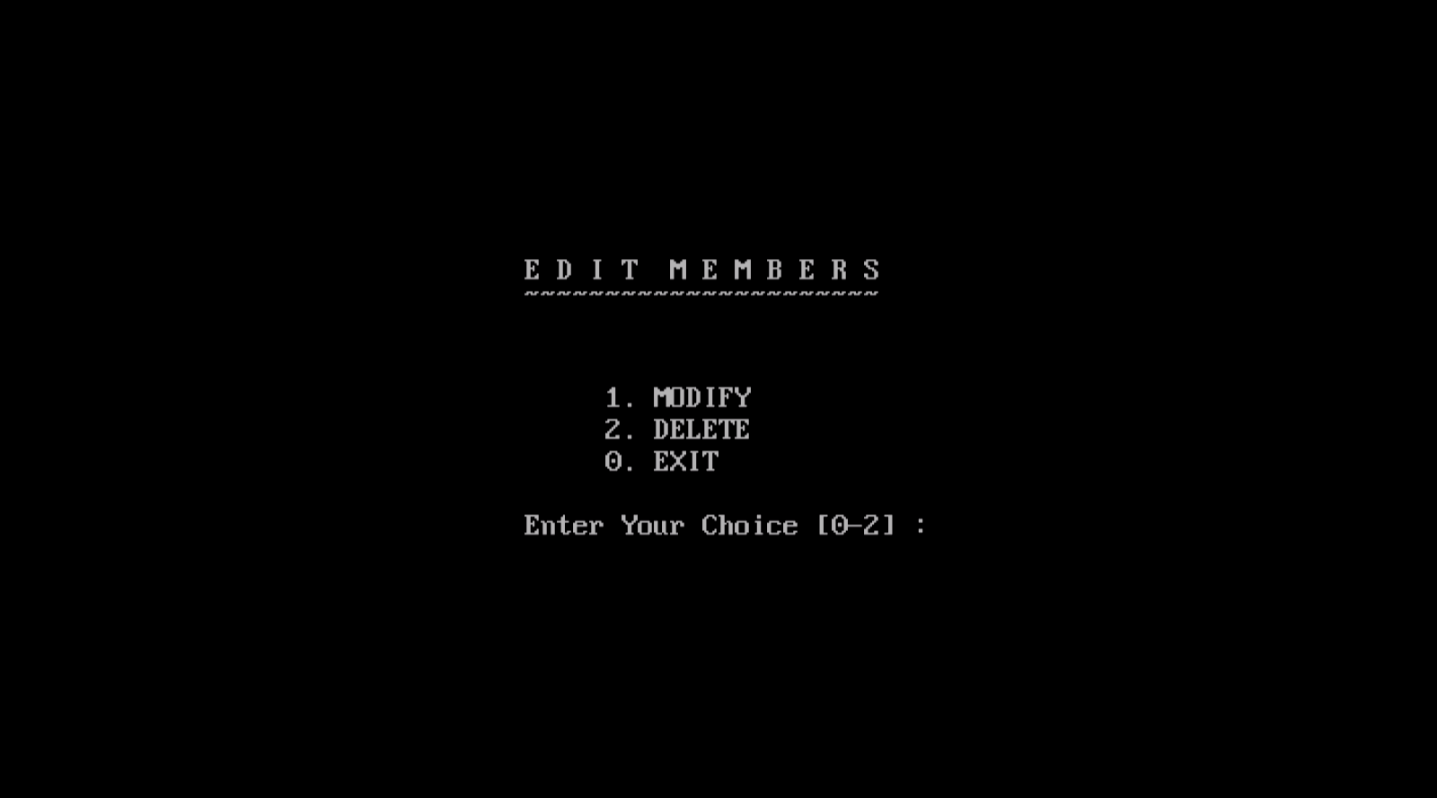


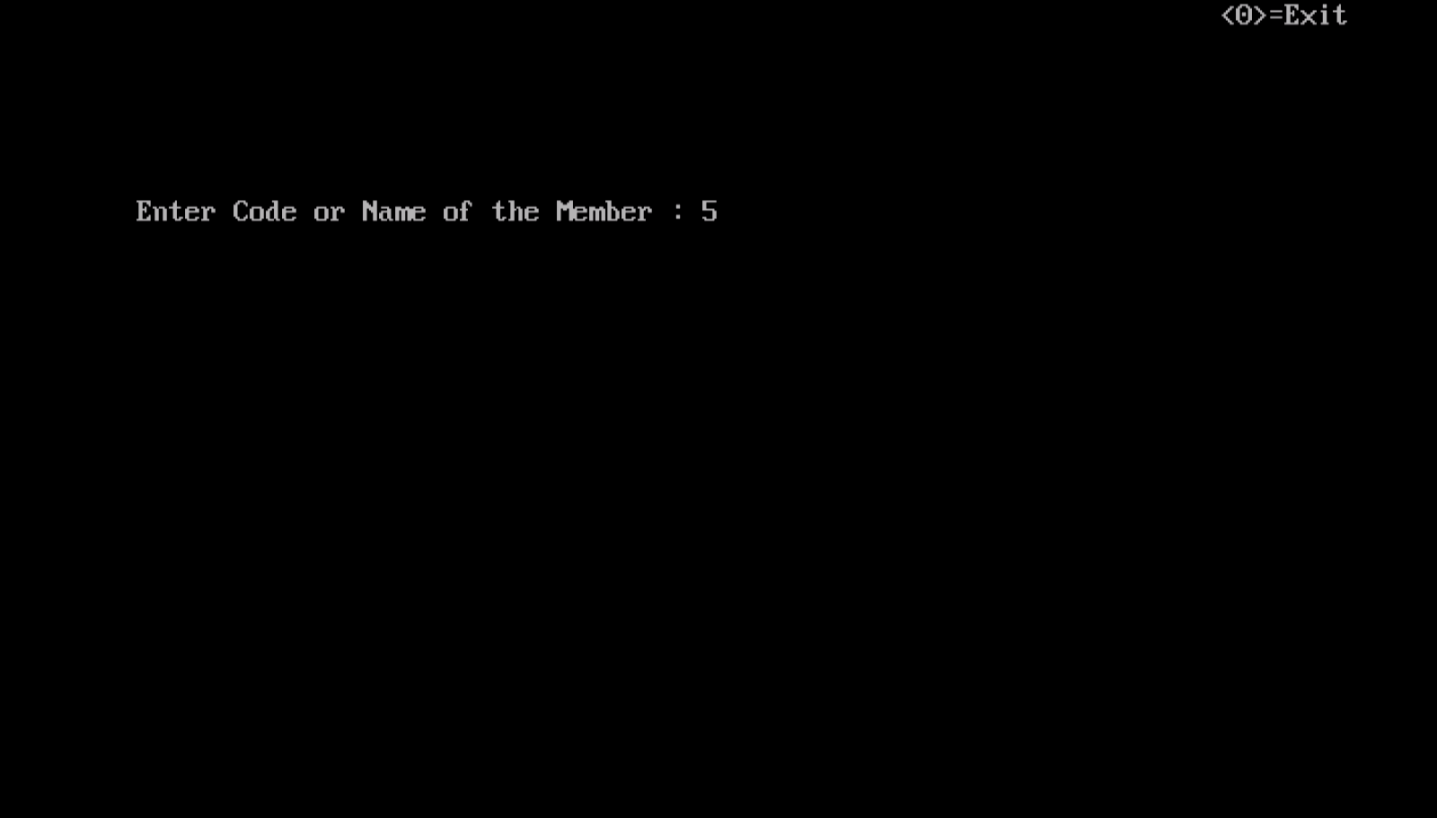
**When NAME of the book to be deleted is typed:-**

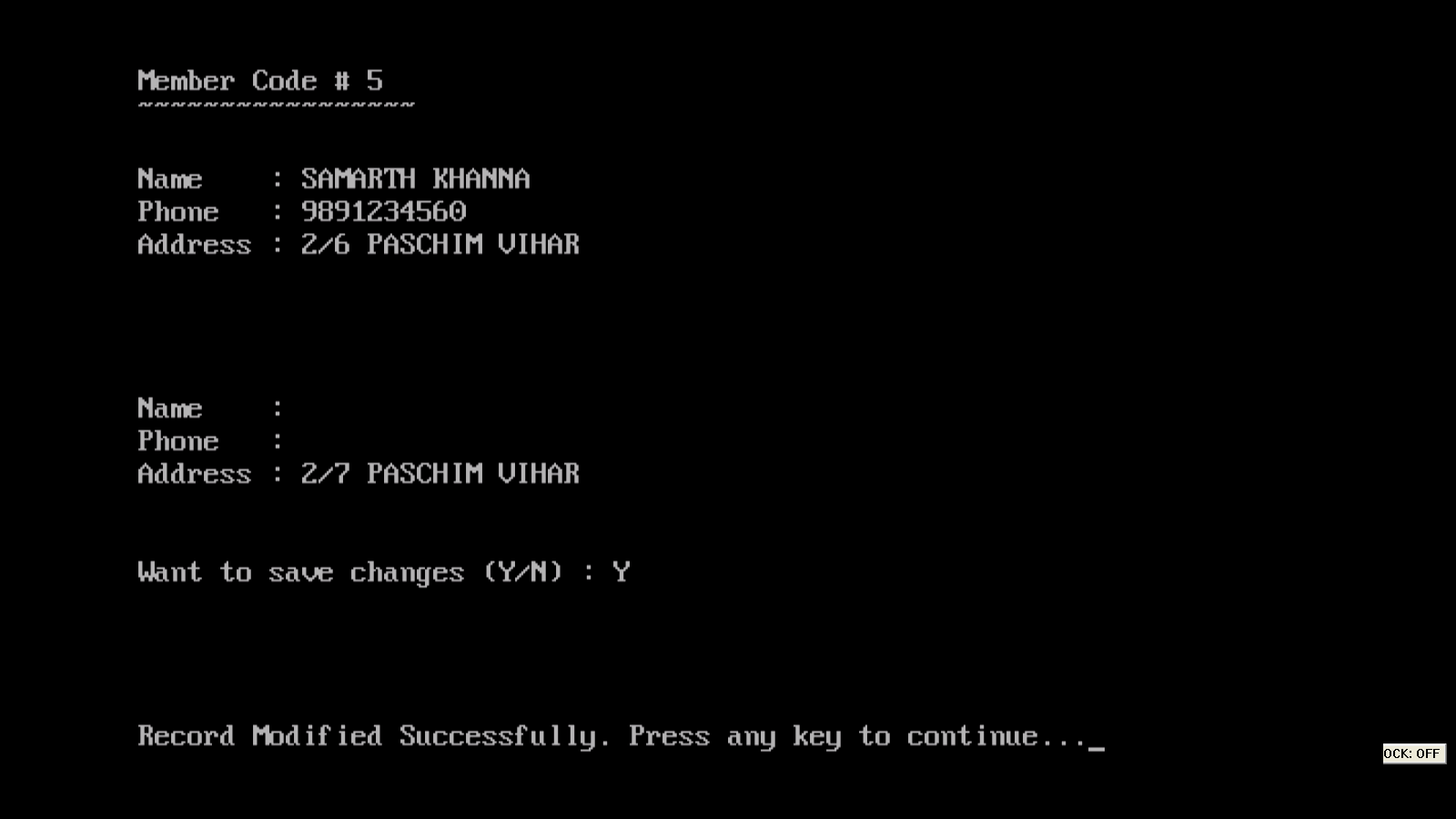


**EDIT MENU**

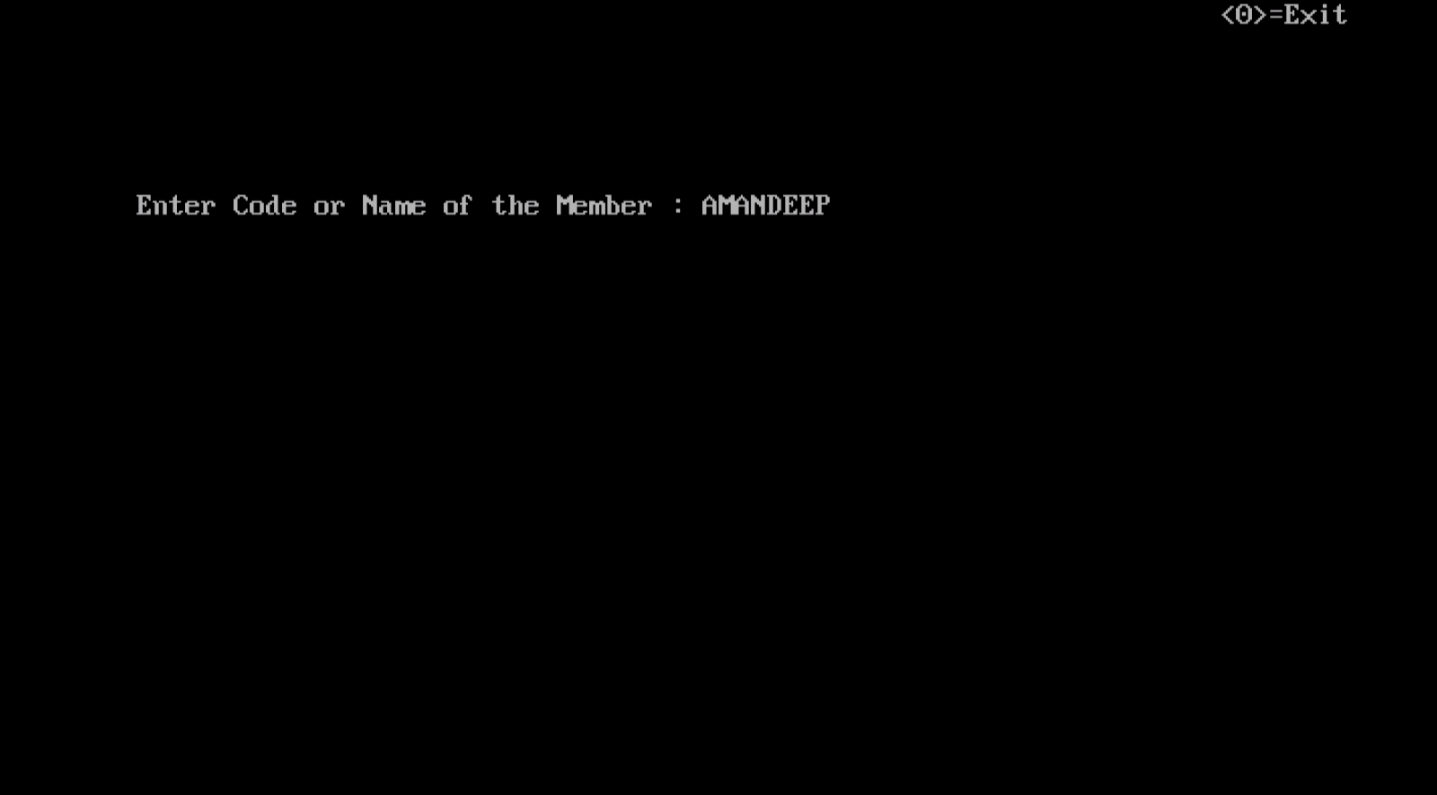
**When option ‘2’ (MEMBERS) is typed:-**

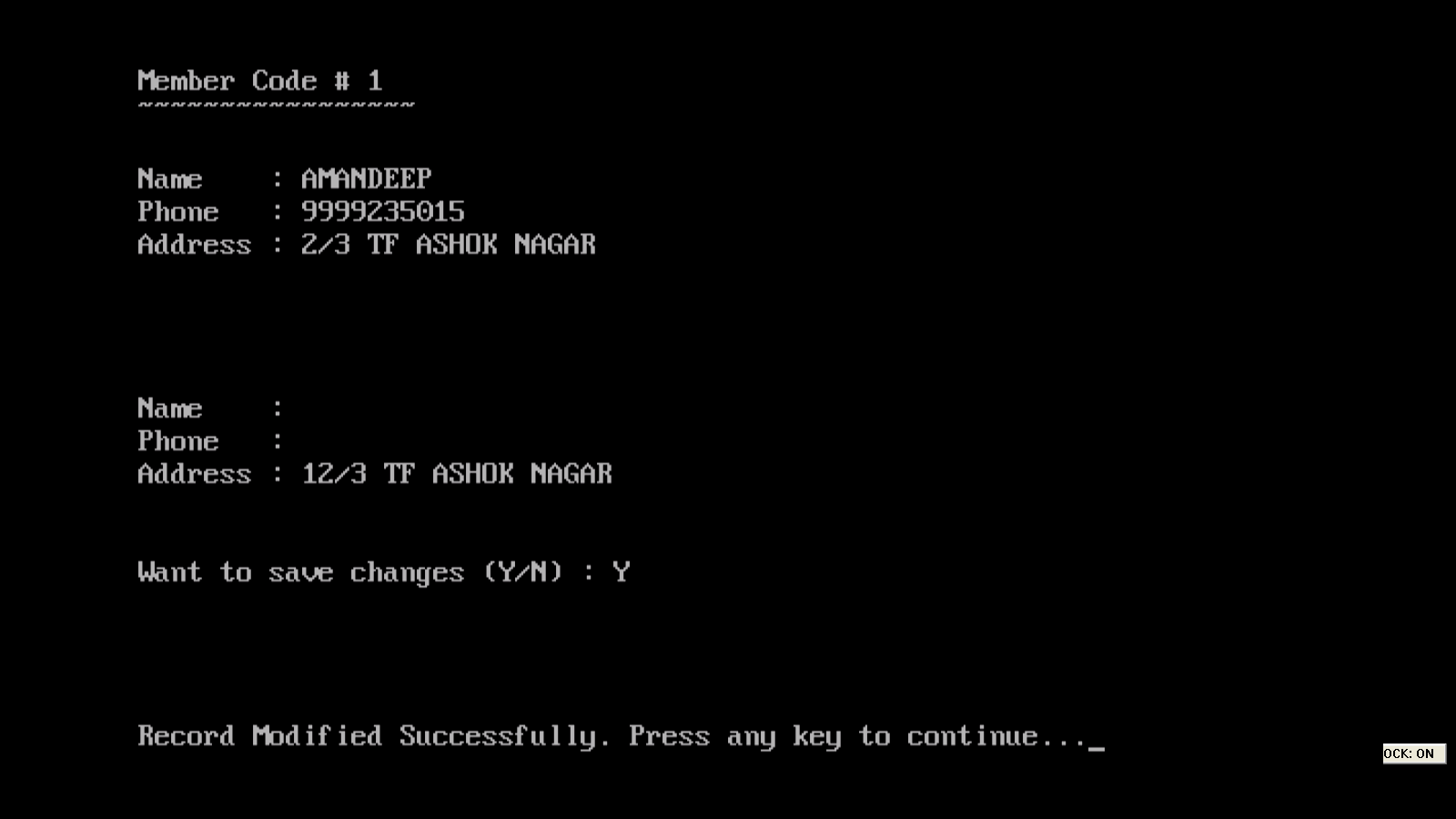


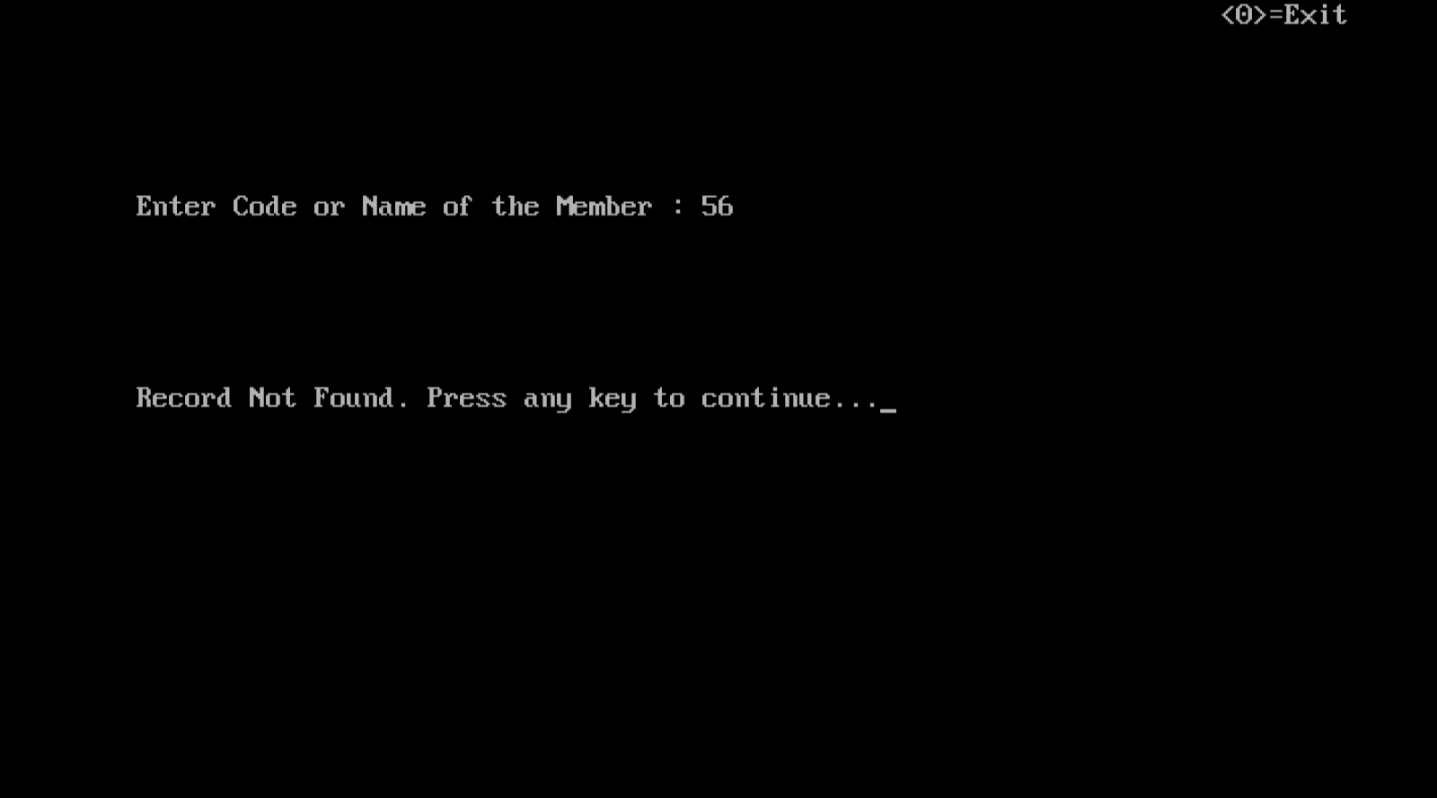
**When CODE of Member whose details is to be modified is typed:-**

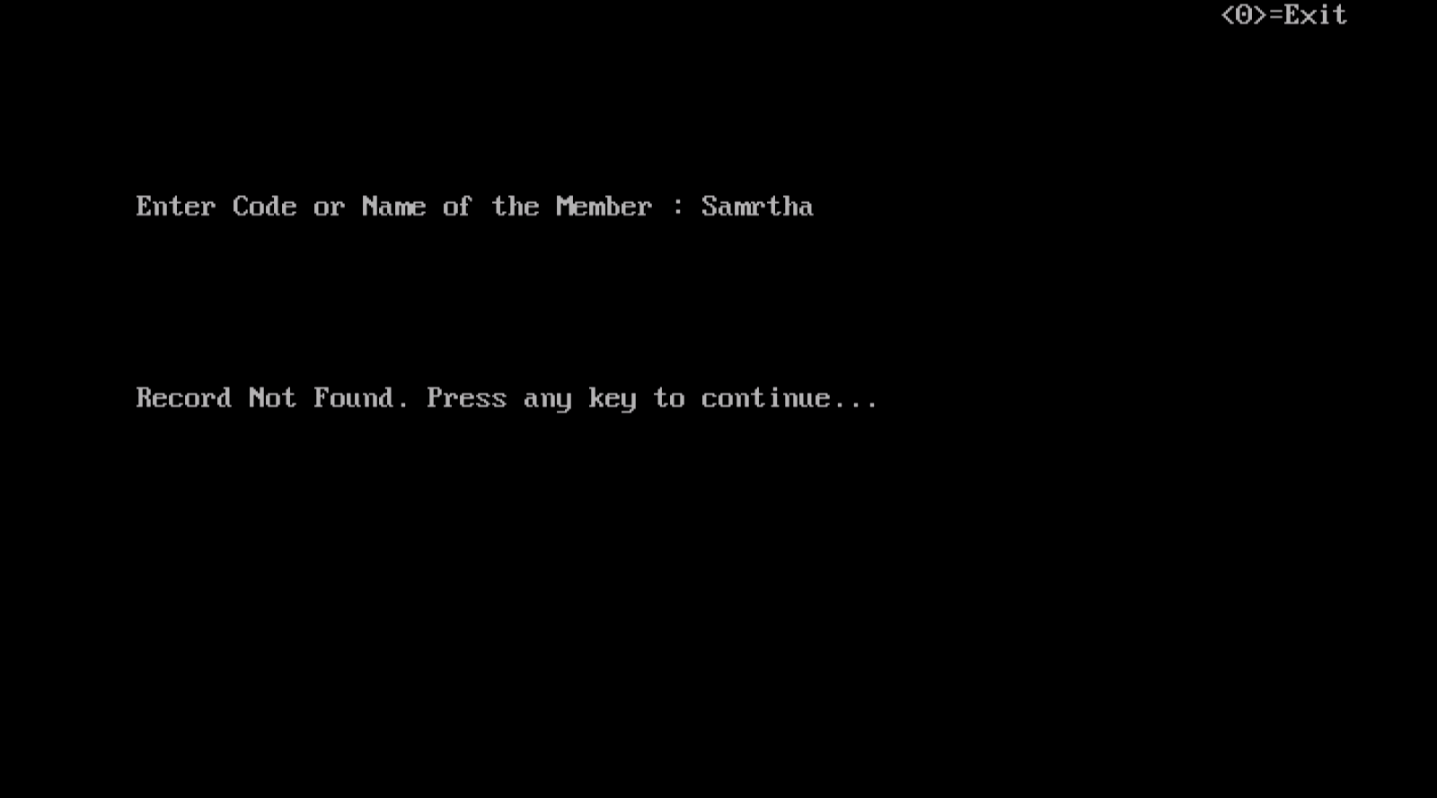


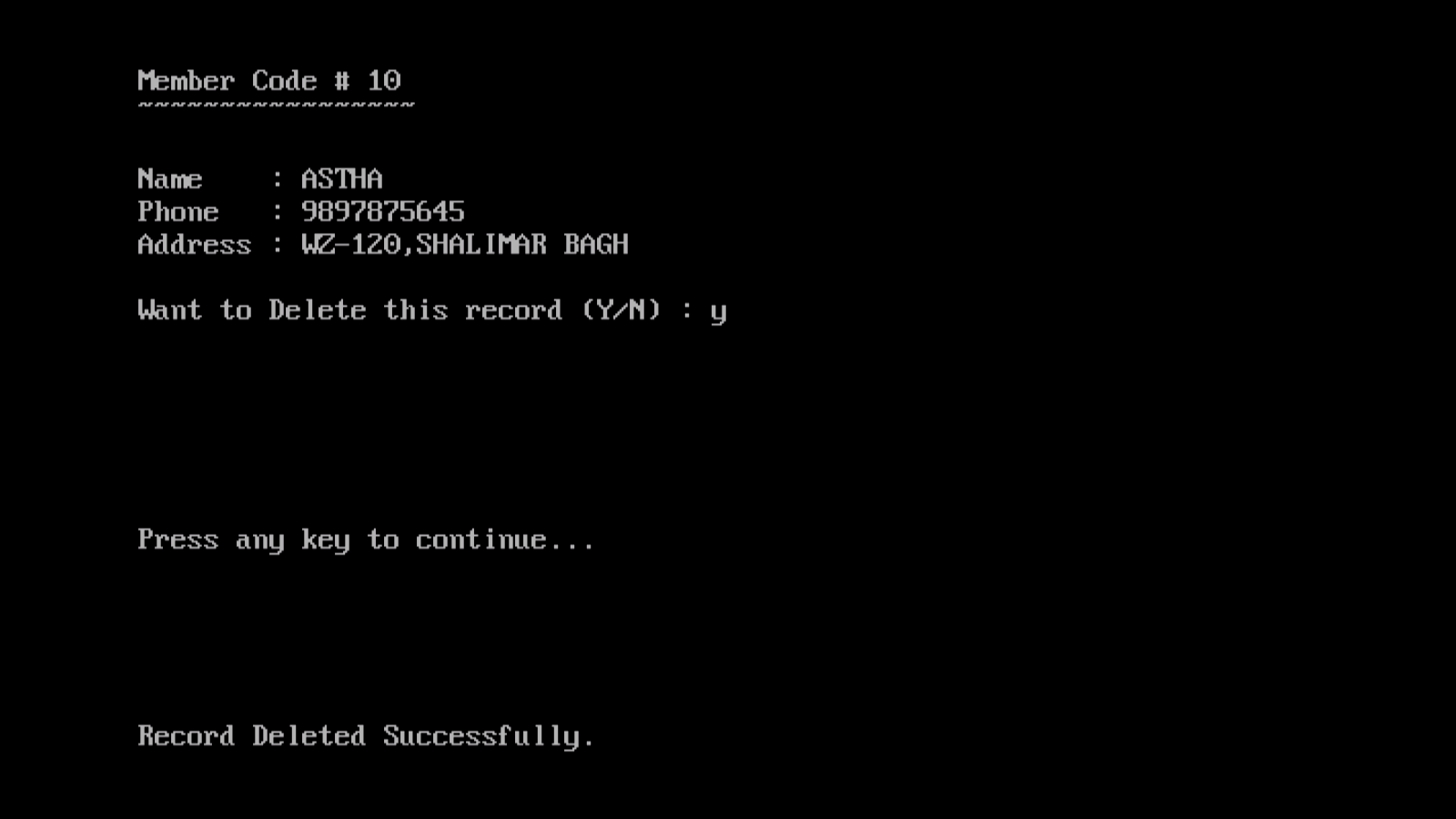
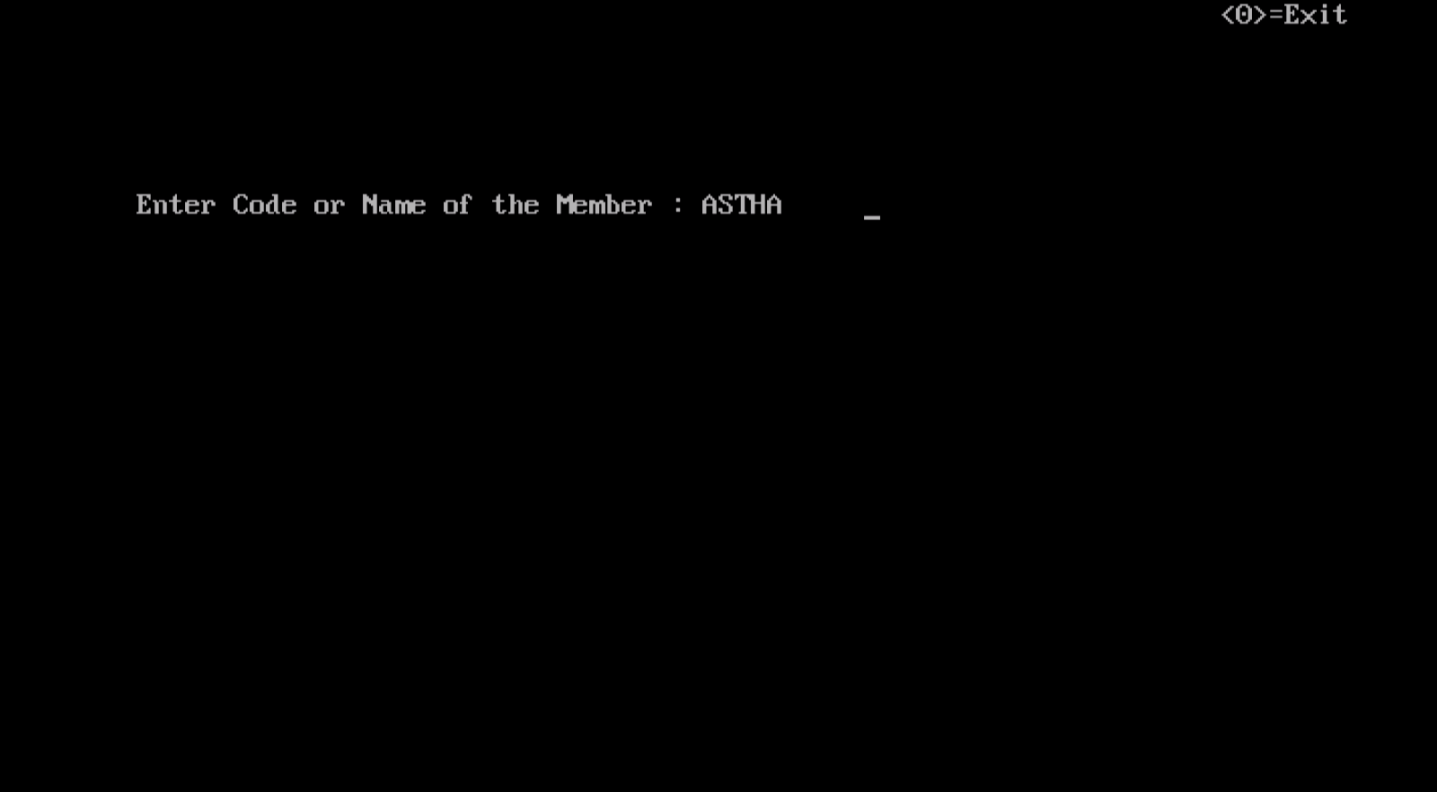
**When NAME of Member whose details are to be modified is typed:-**

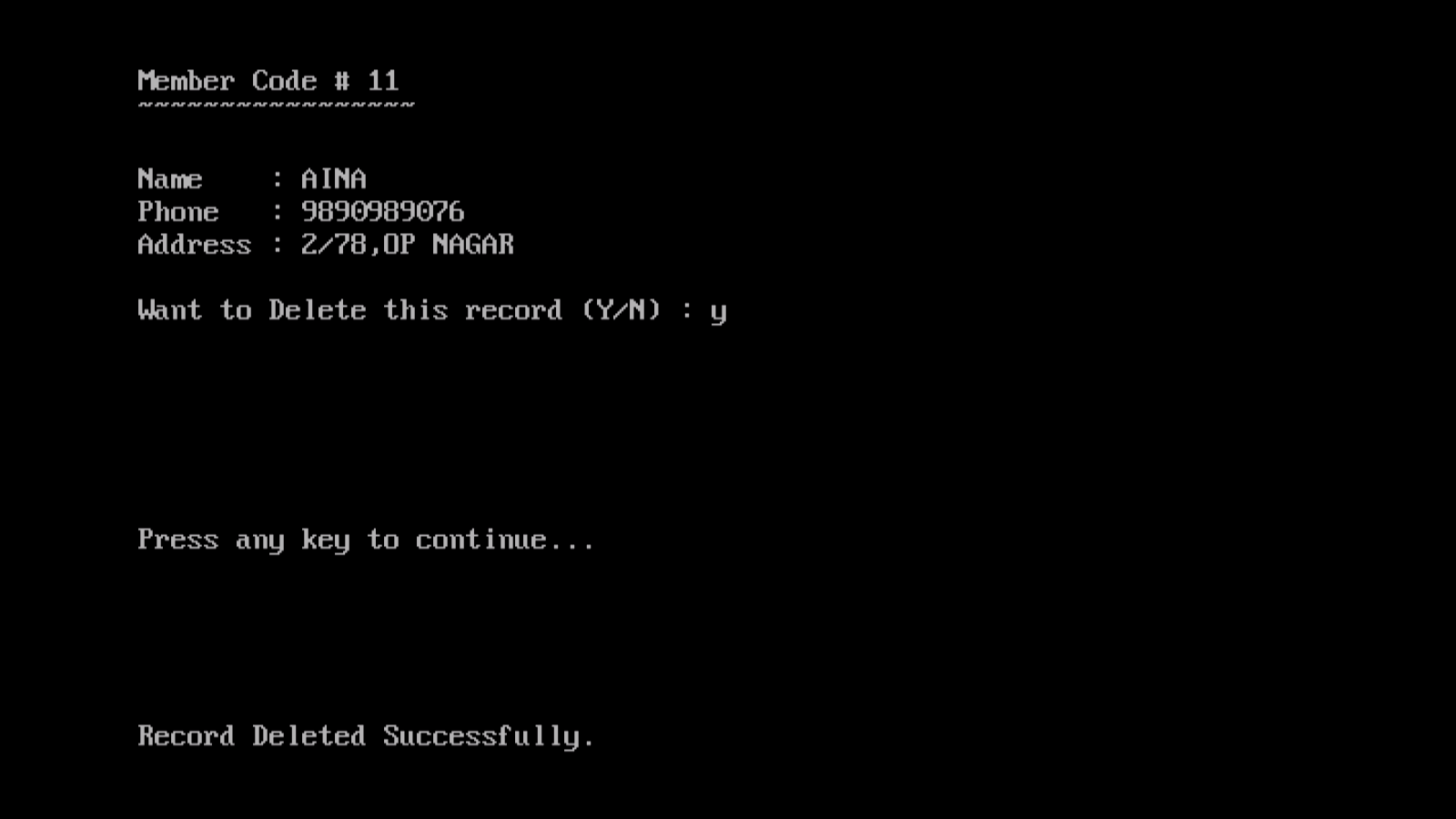
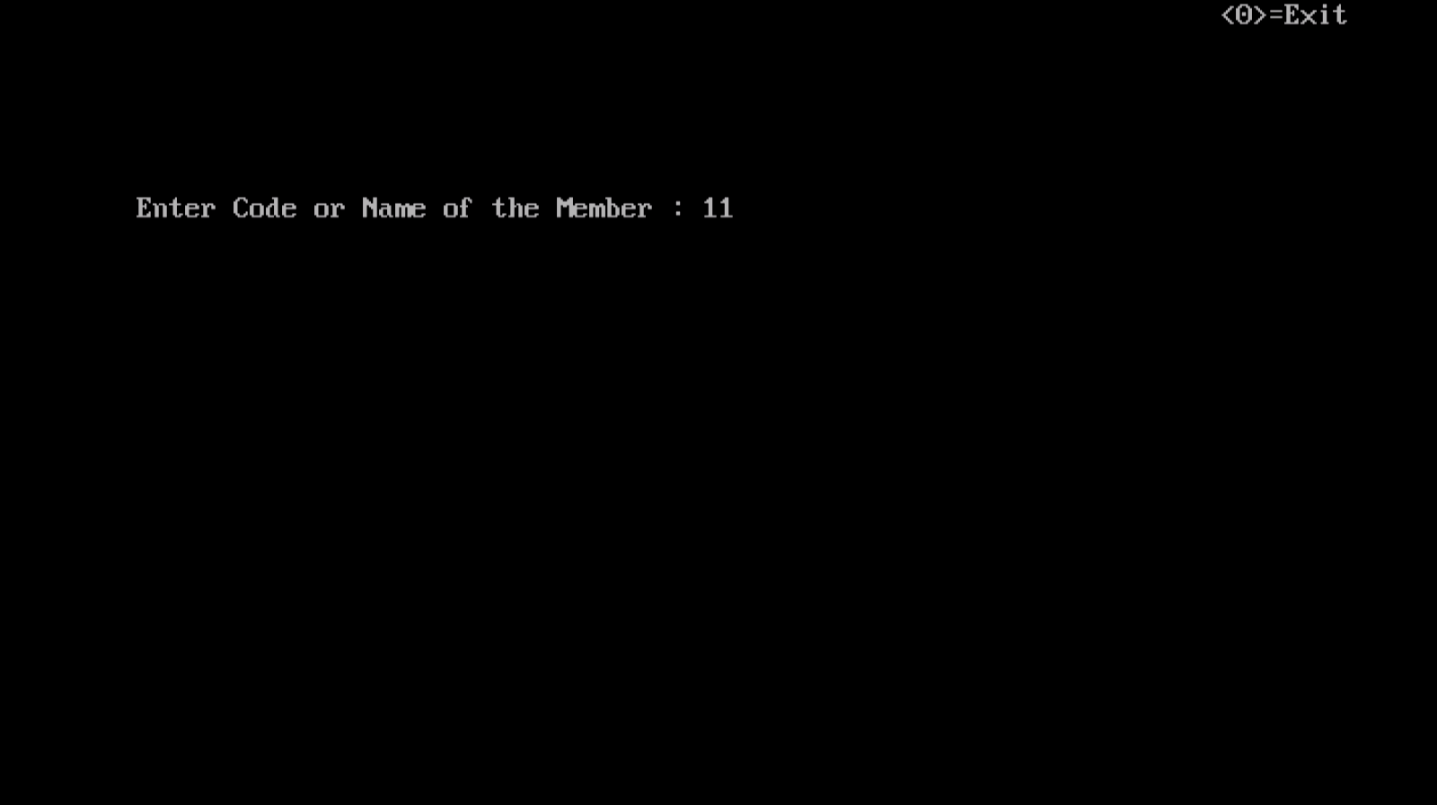


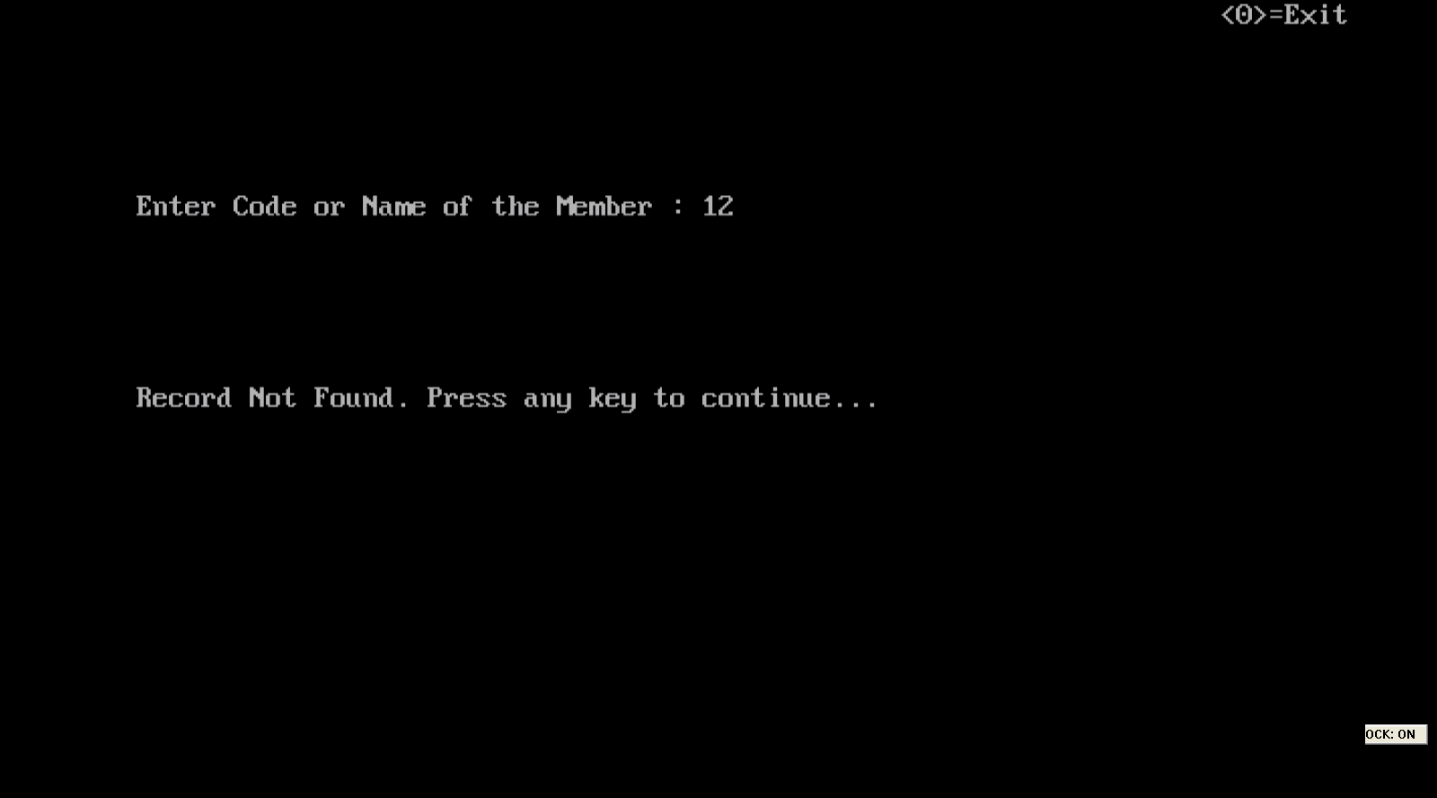


**When INVALID Code of Member is typed:-**

**When INVALID Name of Member is typed:-**

**When option ‘2’(DELETE) Is typed:-**

**When Code of member whose record is to be deleted is typed:-**

**When INVALID Code of Member is typed:-**

**bibliography**

**SOURCES USED:**

* Computer Science textbook by Sumita Arora (class 11 and 12)
* Google.com
* Fundamentals in data structures
* Fundamentals of Computer by Mr.Raja Raman , Printer Hall of India
* Introduction to Computer by Peter Norton , Tata Macgraw hill
* Digital Computer Fundamentals by Thomas C Bartee , Macgraw hills
* Object oriented Programming on C++ by Robert Lafore , Galgotia Publication
* C++ Programming Language by Bjarne Stroustope
* Fundamental of Data Structures by Sartaj & Sahni , Galgota Publications

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*