04/11/2019

|  |
| --- |
|  |

|  |  |
| --- | --- |
| PROJECT | BOOK MANAGEMENT SYSTEM |



**Name –** RITESH TIWARI

**Roll No -** S02201916

**Course –** CSE

**Project Name-** Book Management System

**Software use**

Turbo C++, Snipping Tool, MS Word, Windows 10

**Hardware use**

Mouse, Keyboard, printer, CD;

Contents

[Acknowledgement 4](#_Toc23750963)

[Book Management System 5](#_Toc23750964)

[MAIN MENU 6](#_Toc23750965)

[INSERT MENU 8](#_Toc23750966)

[SEARCH MENU 9](#_Toc23750967)

[DEELETE MENU 11](#_Toc23750968)

[CREATE BOOKS RECORDS 13](#_Toc23750969)

[BOOK RECORDS 15](#_Toc23750970)

[SEARCH BY BOOK ID 16](#_Toc23750971)

[SEARCH BY BOOK NAME 18](#_Toc23750972)

[INSERT FIRST POSITION 20](#_Toc23750973)

[INSERT INTO MIDDLE POSITION 22](#_Toc23750974)

[INSERT INTO LAST POSITION 24](#_Toc23750975)

[DELETE FIRST RECORD 26](#_Toc23750976)

[DELETE LAST RECORD 27](#_Toc23750977)

[DELETE MIDDLE RECORDS 28](#_Toc23750978)

# Acknowledgement

I would like to thank my teacher Anand sir a lot for giving me this opportunity to create such an educational project. This project **(Book Management System)** has given me a lot to learn about **Data Structure in C Language**. I would also like to thank my parents heartily because without their help this project could not be successful. I hope to make such an enlightening project in future also.

Thank you

# Book Management System

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<dos.h>

#include<string.h>

typedef struct book

{

int book\_id;

char book\_name [100];

char author\_name[100];

char publication[100];

char edition [100];

char date\_issue [100];

int isbn\_no;

struct book \*next;

}b;

b \*start;

void create (b \*);

void display (b \*);

b \*insert\_1st();

void insert\_middle ();

void insert\_last ();

b \*delete\_1st ();

void delete\_middle ();

void delete\_last ();

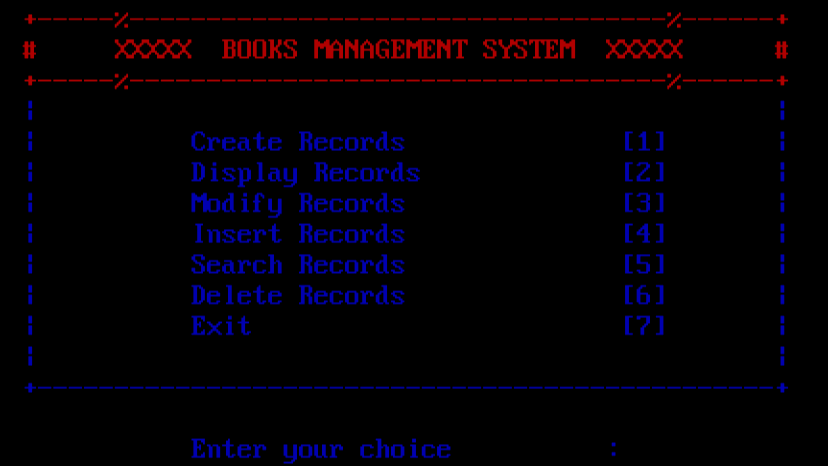
void modify ();

void search\_book\_id ();

void search\_book\_name();

void display1(b \*);

## MAIN MENU



void main(){

int a; clrscr();

start=(b \*)malloc(sizeof(b));

do{

clrscr();

textcolor(RED); printf("\n\t\t");

cprintf("+-----%-----------------------------------%------+"); printf("\n\t\t");

cprintf("# XXXXX BOOKS MANAGEMENT SYSTEM XXXXX # ");

printf("\n\t\t"); cprintf("+-----%-----------------------------------%------+");

textcolor(BLUE);

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("| Create Records [1] |");

printf("\n\t\t"); cprintf("| Display Records [2] |");

printf("\n\t\t"); cprintf("| Modify Records [3] |");

printf("\n\t\t"); cprintf("| Insert Records [4] |");

printf("\n\t\t"); cprintf("| Search Records [5] |");

printf("\n\t\t"); cprintf("| Delete Records [6] |");

printf("\n\t\t"); cprintf("| Exit [7] |");

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("+------------------------------------------------+");

printf("\n\n\t\t");cprintf(" Enter your choice :");

cscanf("%d",&a);

clrscr();

switch(a)

{

case 1: create(start); break;

case 2: display(start); break;

//case 3: modify(); break;

case 4:

INSERT MENU

clrscr();

textcolor(RED);

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

cprintf("+------%-----------------------------------%----+");

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

cprintf("# XXXXX INSERT MENU XXXXX #");

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

cprintf("+------%-----------------------------------%----+");

textcolor(BLUE);

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("| $ Insert in First Position [1] |");

printf("\n\t\t"); cprintf("| $ Insert in Middle Position [2] |");

printf("\n\t\t"); cprintf("| $ Insert in Last Position [3] |");

printf("\n\t\t"); cprintf("| $ Go Back [4] |");

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("+------%-----------------------------------%---+");

printf("\n\n\t\t $ Enter your choice :");

scanf("%d",&a);

switch(a)

{

case 1: clrscr(); start=insert\_1st(); break;

case 2: clrscr(); insert\_middle(); break;

case 3: clrscr(); insert\_last(); break;

case 4: clrscr(); break;

default : printf("\n You pressed the wrong option, please try again!\n");

}

break;

case 5:

SEARCH MENU

clrscr();

textcolor(RED);

printf("\n\t\t"); cprintf("+------%-----------------------------------%---+");

printf("\n\t\t"); cprintf(" # XXXXX SEARCH MENU XXXXX #");

printf("\n\t\t"); cprintf("+------%-----------------------------------%---+");

textcolor(BLUE);

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("| $ Search By Book ID [1] $ |");

printf("\n\t\t"); cprintf("| $ Search by Book Name [2] $ |");

printf("\n\t\t"); cprintf("| $ Go back [3] $ |");

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("+------%-----------------------------------%---+");

printf("\n\t\t $ Enter your choice :");

scanf("%d",&a);

switch(a)

{

case 1: clrscr();

search\_book\_id();

break;

case 2: clrscr();

search\_book\_name();

break;

case 3: clrscr();

break;

default: printf("\n You pressed the wrong option, please try again!\n");

}

break;

case 6:

DEELETE MENU

clrscr();

textcolor(RED);

printf("\n\t\t"); cprintf("+-------%-------------------------------------------%----+");

printf("\n\t\t"); cprintf("# XXXXX DELETE MENU XXXXX #");

printf("\n\t\t"); cprintf("+-------%-------------------------------------------%----+");

textcolor(BLUE);

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("| $ Delete from First Position [1] $ |");

printf("\n\t\t"); cprintf("| $ Delete from Middle Position [2] $ |");

printf("\n\t\t"); cprintf("| $ Delete from Last Position [3] $ |");

printf("\n\t\t"); cprintf("| $ Go Back [4] $ |");

printf("\n\t\t"); cprintf("| |");

printf("\n\t\t"); cprintf("+-------%-------------------------------------------%----+");

printf("\n\n\t\t $ Enter your choice :");

scanf("%d",&a);

switch(a)

{

case 1: clrscr();

start=delete\_1st();

break;

case 2: clrscr();

delete\_middle();

break;

case 3: clrscr();

delete\_last();

break;

case 4: clrscr();

break;

default: printf("\nYou pressed the wrong option, please try again!\n");

}

break;

case 7:

exit(0);

default:

printf("\n You pressed the wrong option, please try again!\n");

}

}while(1);

}

## CREATE BOOKS RECORDS



void create(b \*node)

{

char ans='y';

while(1)

{

clrscr();

textcolor(YELLOW);

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

cprintf("$ XXXXX CREATE BOOKS RECORDS XXXXX $");

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

textcolor(WHITE);

node->next=(b \*)malloc(sizeof(b));

cprintf(" \* Book ID :"); scanf("%d",&node->book\_id);

printf("\n\t\t"); cprintf(" \* Book Name :");

fflush(stdin); gets(node->book\_name);

printf("\n\t\t"); cprintf(" \* Author Name :");

fflush(stdin); gets(node-> author\_name);

printf("\n\t\t"); cprintf(" \* Publication :");

fflush(stdin); gets(node->publication);

printf("\n\t\t"); cprintf(" \* Edition :");

fflush(stdin); gets(node->edition);

printf("\n\t\t"); cprintf(" \* Date Issue :");

fflush(stdin); gets(node->date\_issue);

printf("\n\t\t"); cprintf(" \* ISBN NO :");

scanf("%d",&node->isbn\_no);

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

textcolor(YELLOW);

cprintf("$ $");

printf("\n\n\t\t"); cprintf(" Do you want to continue [Y/N]:");

fflush(stdin); ans=getchar();

if(ans=='y' || ans=='Y')

{ node=node->next; }

else { node->next='\0'; break; }

}

}

## BOOK RECORDS



void display(b \*node)

{

int no=0;

char n;

clrscr();

printf("\n\t\t\t\*\*\*\*\*BOOK RECORDS\*\*\*\*\*\n\n\t");

textcolor(RED);

cprintf("|Book ID Book Name Author Name Publication Issue Date |\n");

while(node)

{

textcolor(WHITE);

printf("\n\t");

cprintf("| %d %s %s %s %s |",node->book\_id,node->book\_name,node->author\_name,node->publication,node->date\_issue);

node=node->next;

no++;

}

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

cprintf("Do you want to see the details [Y/N] :");

fflush(stdin);

n=getchar();

if(n=='y'||n=='Y')

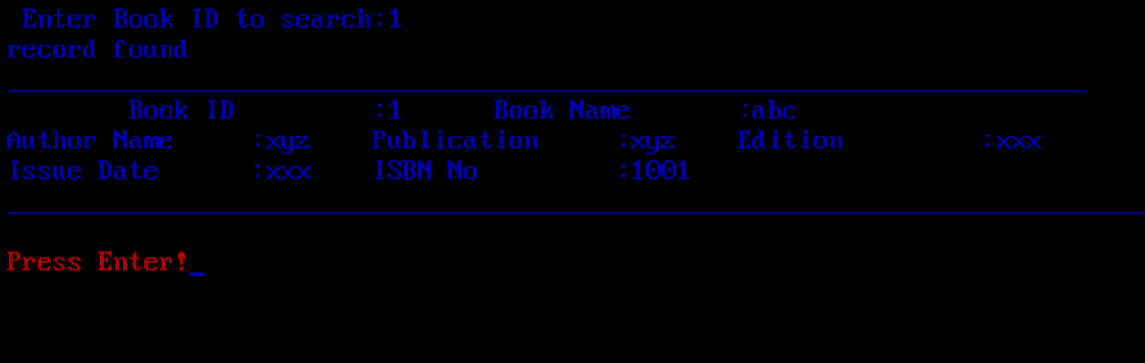
{ printf("\n\t\t\t"); search\_book\_id(); return; }

textcolor(RED+BLINK); printf("\n");

cprintf("Press Enter!") getch();

}

## SEARCH BY BOOK ID



void search\_book\_id()

{

b \*node,\*temp;

int nod;

node=start;

temp=start;

printf("\n Enter Book ID to search:");

scanf("%d",&nod);

while(node)

{

if(temp->book\_id==nod)

{

printf("record found");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\tBook ID \t:%d",node->book\_id);

printf("\tBook Name \t:%s\n",node->book\_name);

printf("Author Name\t:%s",node->author\_name);

printf("\tPublication \t:%s",node->publication);

printf("\tEdition\t\t:%s\n",node->edition);

printf("\Issue Date \t:%s",node->date\_issue);

printf("\tISBN No\t\t:%d",node->isbn\_no);

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

getch();

break;

}

else { printf("Record Not found"); break; }

node=node->next;

}

textcolor(RED+BLINK);

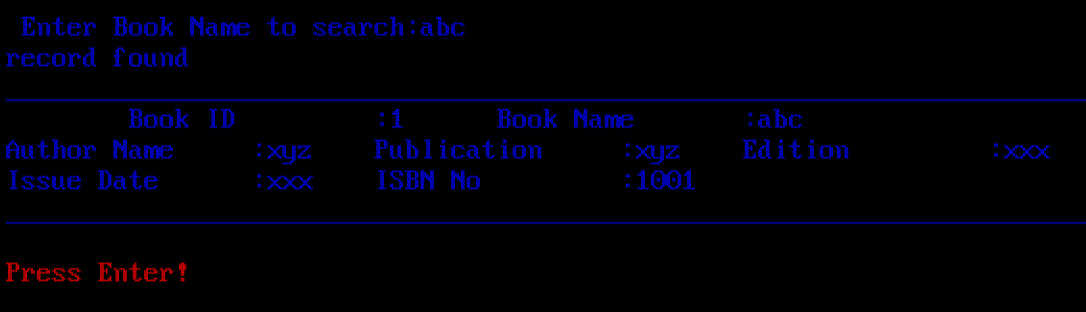
printf("\n");

cprintf("Press Enter!");

getch();

}

## SEARCH BY BOOK NAME



void search\_book\_name()

{

b \*node;

char name[100];

node=start;

printf("\n Enter Book Name to search:");

fflush(stdin);

gets(name);

while(node)

{

if(strcmp(node->book\_name,name)==0)

{

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\tBook ID \t:%d",node->book\_id);

printf("\tBook Name \t:%s\n",node->book\_name);

printf("Author Name\t:%s",node->author\_name);

printf("\tPublication \t:%s",node->publication);

printf("\tEdition\t\t:%s\n",node->edition);

printf("\tIssue Date \t:%s",node->date\_issue);

printf("\tISBN No\t\t:%d",node->isbn\_no);

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

getch();

break;

}

else

{

printf("Record Not found");

break;

}

node=node->next;

}

textcolor(RED+BLINK);

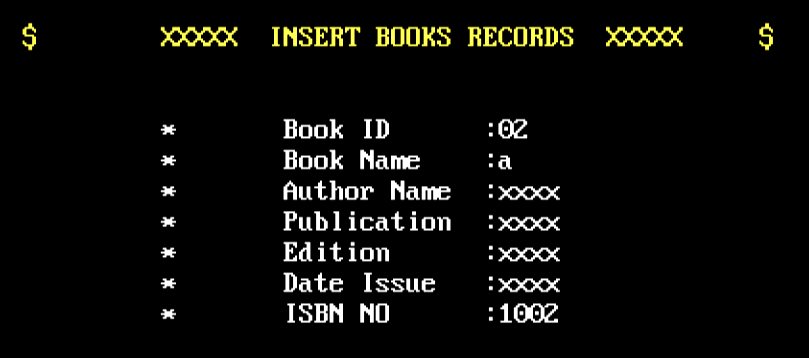
printf("\n");

cprintf("Press Enter!");

getch();

}

## INSERT FIRST POSITION



b \*insert\_1st()//b \*insert\_1st()

{

b \*node,\*new1;

node=start;

new1=(b \*)malloc(sizeof(b));

new1->next=node;

textcolor(YELLOW);

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

cprintf("$ XXXXX INSERT BOOKS RECORDS XXXXX $");

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

textcolor(WHITE);

cprintf(" \* Book ID :"); cscanf("%d",&new1->book\_id);

printf("\n\t\t"); cprintf(" \* Book Name :");

fflush(stdin); gets(new1->book\_name);

printf("\n\t\t"); cprintf(" \* Author Name :");

fflush(stdin); gets(new1->author\_name);

printf("\n\t\t"); cprintf(" \* Publication :");

fflush(stdin); gets(new1->publication);

printf("\n\t\t"); cprintf(" \* Edition :");

fflush(stdin); gets(new1->edition);

printf("\n\t\t"); cprintf(" \* Date Issue :");

fflush(stdin); gets(new1->date\_issue);

printf("\n\t\t"); cprintf(" \* ISBN NO :");

cscanf("%d",&new1->isbn\_no);

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

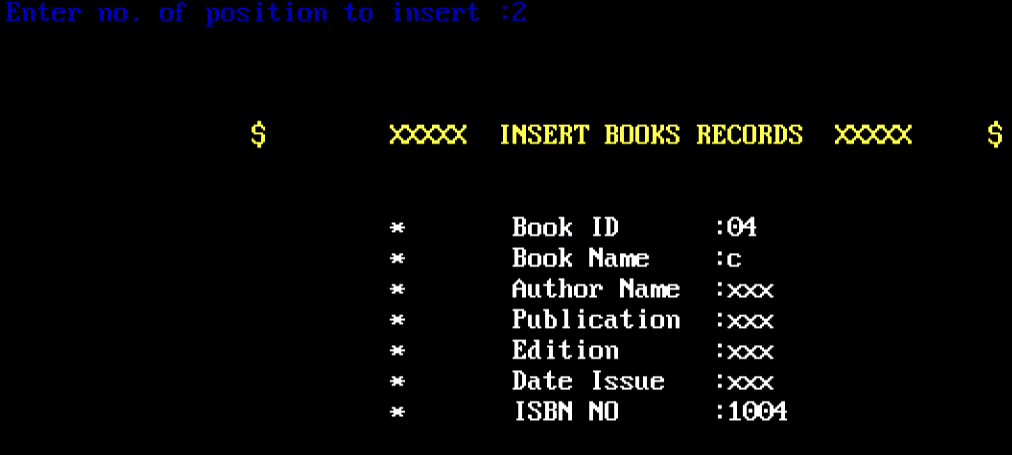
textcolor(YELLOW);

cprintf("$ XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX $");

start=new1;

return start;

}

INSERT INTO MIDDLE POSITION 

void insert\_middle()

{

b \*prev,\*node,\*new1;

int no=0,p=0;

prev=start;

node=start->next;

printf("Enter no. of position to insert :");

scanf("%d",&p);

while(node)

{

no++;

if(no==p)

{

new1=(b \*)malloc(sizeof(b));

textcolor(YELLOW);

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

cprintf("$ XXXXX INSERT BOOKS RECORDS XXXXX $");

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

textcolor(WHITE);

cprintf(" \* Book ID :");

cscanf("%d",&new1->book\_id);

printf("\n\t\t"); cprintf(" \* Book Name :");

fflush(stdin); gets(new1->book\_name);

printf("\n\t\t"); cprintf(" \* Author Name :");

fflush(stdin); gets(new1->author\_name);

printf("\n\t\t"); cprintf(" \* Publication :");

fflush(stdin); gets(new1->publication);

printf("\n\t\t"); cprintf(" \* Edition :");

fflush(stdin); gets(new1->edition);

printf("\n\t\t"); cprintf(" \* Date Issue :");

fflush(stdin); gets(new1->date\_issue);

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

cprintf(" \* ISBN NO :");

cscanf("%d",&new1->isbn\_no);

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

textcolor(YELLOW);

cprintf("$ XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX $");

new1->next=node;

prev->next=new1;

break;

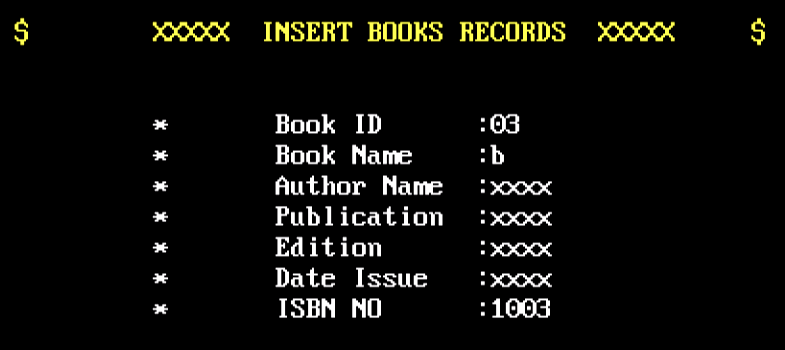
}

else { prev=prev->next; node=node->next; }

}

}

## INSERT INTO LAST POSITION



void insert\_last()

{

b \*node,\*new1,\*prev;

prev=start;

node=start->next;

while(node)

{ node=node->next; prev=prev->next; }

if(node==NULL)

{

new1=(b \*)malloc(sizeof(b));

new1->next=node;

prev->next=new1;

textcolor(YELLOW);

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

cprintf("$ XXXXX INSERT BOOKS RECORDS XXXXX $");

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

textcolor(WHITE);

cprintf(" \* Book ID :");

cscanf("%d",&new1->book\_id);

printf("\n\t\t"); cprintf(" \* Book Name :");

fflush(stdin); gets(new1->book\_name);

printf("\n\t\t"); cprintf(" \* Author Name :");

fflush(stdin); gets(new1->author\_name);

printf("\n\t\t"); cprintf(" \* Publication :");

fflush(stdin); gets(new1->publication);

printf("\n\t\t"); cprintf(" \* Edition :");

fflush(stdin); gets(new1->edition);

printf("\n\t\t"); cprintf(" \* Date Issue :");

fflush(stdin); gets(new1->date\_issue);

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

cprintf(" \* ISBN NO :");

cscanf("%d",&new1->isbn\_no);

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

textcolor(YELLOW);

cprintf("$ XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX $");

}

}

## DELETE FIRST RECORD



b \*delete\_1st()

{

b \*node;

clrscr();

node=start;

start=node->next;

free(node);

textcolor(YELLOW);

cprintf(“\t\t”);

printf(" 1st Position record was deleted...");

display(start);

getch();

return start;

}

## DELETE LAST RECORD



void delete\_last()

{

b \*prev, \*node;

clrscr();

node=start;

prev=start;

while(node)

{

if(node->next==NULL)

{ prev->next=node->next; free(node); break; }

else { prev=node; node=node->next; }

}

textcolor(YELLOW);

printf(“\t\t”);

cprintf("Last record was deleted ...");

display(start);

getch();

}

## DELETE MIDDLE RECORDS



void delete\_middle()

{

b \*prev ,\*node;

int no=1,nod=0;

clrscr();

prev=start;

node=start->next;

printf("Enter book id to delete:");

scanf("%d",&nod);

while(node)

{

if(node->book\_id==nod)

{ prev->next=node->next; free(node); break; }

prev=prev->next; node=node->next; no++;

}

textcolor(YELLOW);

printf(“\t\t”);

cprintf("%d was deleted",nod);

display(start);

getch();

}