| **Team No.** | **Team Member names with USNs** |
| --- | --- |
| 6 | 1.Veeresh Kumar Y M, 1SI20CS126  2.Yukthish R, 1SI20CS134  3.Venkatesha S, 1SI20CS127  4.Shadakshari, 1SI20CS102 |
| **Question** | Consider a singly linked list with each node containing following information about a book: title, author, pages and price. Develop a C program using functions to perform the following operations on this list: a) Add a book at the beginning of the list. b) Delete a book of specified author. c) List all the books whose price is in the range Rs. 1000-Rs.2000. d) List the details of the books whose title starts with letter “C”. |

**Solution:**

#include <stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct node //daclearing a structure

{

char author[50], title[50];

int page;

float price;

struct node \*next; //self declearing structure

int info;

} NODE;

void Insert(NODE \*head)

{

NODE \*newnode;

newnode=(NODE\*)malloc(sizeof(NODE));

printf("\nEnter the title of the book:");

scanf("%s",newnode->title);

printf("\nEnter the author name:");

scanf("%s",newnode->author);

printf("\nEnter the page no:");

scanf("%d",&newnode->page);

printf("\nEnter the book price:");

scanf("%f",&newnode->price);

newnode->next=head->next;

head->next=newnode;

head->info++;

printf("\nBook %s is added at the beginning of the list",newnode->title);

}

void Delete(NODE \*head)

{

NODE \*temp,\*prev;

char author[50];

if(head->next==NULL)

{

printf("\nEMPTY LIST\n");

return;

}

else

{

printf("\nEnter the author name:");

scanf("%s",author);

temp=head->next;

prev=head;

while(temp!=NULL && strcmp(temp->author,author)!=0)

{

prev=prev->next;

temp=temp->next;

}

if(temp==NULL)

{

printf("\nAuthor is not found\n");

}

else

{

prev->next=temp->next;

free(temp);

printf("\nBook by author %s is deleted",author);

}

}

}

void Display1(NODE \*head)

{

NODE \*temp;

int flag=0;

if( head->next==NULL )

{

printf("\nEmpty List");

return;

}

temp=head->next;

while(temp!=NULL)

{

if(temp->price >= 1000.0 && temp->price <=2000.0)

{

if(flag==0)

{

printf("\nBOOK\t\tAUTHOR\tPRICE\tPAGES\n");

flag=1;

}

printf("%s\t\t%s\t%.0f\t%d\n",temp->title,temp->author,temp->price,

temp->page);

}

temp=temp->next;

}

if(flag==0)

printf("\nNo books are found in range 1000 to 2000\n");

}

void Display2(NODE \*head)

{

NODE \*temp;

int flag1=0;

if(head->next==NULL)

{

printf("\nEmpty List");

return;

}

temp=head->next;

while(temp!=NULL)

{

if(temp->title[0]=='c' || temp->title[0]=='C')

{

if(flag1==0)

{

printf("\nBOOK\t\tAUTHOR\tPRICE\tPAGES\n");

flag1++;

}

printf("%s\t\t%s\t%.0f\t%d\n",temp->title,temp->author,temp->price,

temp->page);

}

temp=temp->next;

}

if(flag1==0)

printf("\nNo Books found with starting letter C\n");

}

void Displayall(NODE \*head)

{

NODE \*temp;

if(head->next==NULL)

{

printf("\nEMPTY LIST");

return;

}

temp=head->next;

printf("\nBOOK\t\tAUTHOR\tPRICE\tPAGES\n");

while(temp!=NULL)

{

printf("%s\t\t%s\t%.0f\t%d\n",temp->title,temp->author,temp->price,

temp->page);

temp=temp->next;

}

}

int main()

{

NODE \*head;

head=(NODE\*)malloc(sizeof(NODE));

head->info=0;

head->next=NULL;

int choice;

while(1)

{

printf("\n\n 1:INSERT BOOK\n 2:DELETE BOOK\n 3:DISPLAY1\n

4:DISPLAY2\n 5:DISPLAYALL \n 6:EXIT\n");

printf("\n enter the choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:Insert(head);

break;

case 2:Delete(head);

break;

case 3:Display1(head);

break;

case 4:Display2(head);

break;

case 5:Displayall(head);

break;

case 6:exit(0);

default:printf("\nINVALID CHOICE");

}

}

return 0;

}

**Output:**



