DOUBLY LINKED LIST

PROGRAM:-

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

struct node

{

struct node \*prev;

int no, rate;

struct node \*next;

};

struct node \*start=NULL;

void append(int n);

void insert(int n);

void del(int n);

void display();

void main()

{

int n,r,x,y;

clrscr();

printf("\n1.Add at first\n2.Append\n3.Delete\n4.Display\n5.Exit");

do

{

printf("\nyour choice : ");

scanf("%d",&x);

switch(x)

{

case 1:

printf("\n Menu:-");

printf("\n (1) Ladoo\n (2) Jillaebee\n (3) Milk sweet");

printf("\n Enter choice : ");

scanf("%d",&n);

insert(n);

break;

case 2:

printf("\n Menu:-");

printf("\n (1) Ladoo\n (2) Jillaebee\n (3) Milk sweet");

printf("\n Enter choice : ");

scanf("%d",&n);

append(n);

break;

case 3:

printf("\nEnter the chosen number to delete\n");

scanf("%d",&n);

del(n);

break;

case 4:

display();

break;

default:

x=5;

}

}while(x!=5);

}

void append(int n)

{

struct node \*nptr,\*temp=start;

nptr=(struct node\*)malloc(sizeof(struct node));

switch(n)

{

case 1:

{

nptr->no=n;

nptr->rate=100;

}break;

case 2:

{

nptr->no=n;

nptr->rate=200;

}break;

case 3:

{

nptr->no=n;

nptr->rate=300;

}break;

default:

printf("\n.......error");

}

nptr->next=NULL;

nptr->prev=NULL;

if(start==NULL)

{

start=nptr;

}

else

{

while(temp->next!=NULL)

temp=temp->next;

nptr->prev=temp;

temp->next=nptr;

}

}

void insert(int n)

{

struct node \*nptr;

nptr=(struct node\*)malloc(sizeof(struct node));

nptr->prev=NULL;

nptr->no=n;

switch(n)

{

case 1:

{

nptr->no=n;

nptr->rate=100;

}break;

case 2:

{

nptr->no=n;

nptr->rate=200;

}break;

case 3:

{

nptr->no=n;

nptr->rate=300;

}break;

default:

printf("\n.......error");

}

nptr->next=start;

if(start!=NULL)

{

start->prev=nptr;

}

start=nptr;

}

void del(int n)

{

struct node \*temp=start;

while(temp!=NULL)

{

if(temp->no==n)

{

if(temp==start)

{

start=start->next;

start->prev=0;

}

else

{

if(temp->next==NULL)

temp->prev->next=NULL;

else

{

temp->prev->next=temp->next;

temp->next->prev=temp->prev;

}

free(temp);

}

return;

}

temp=temp->next;

}

printf("%d item not found",n);

}

void display()

{

struct node \*cur=start, \*prev=NULL;

printf("\nForward traversal:\n");

while(cur!=NULL)

{

if(cur->no==1)

{

printf("\n ITEM : Ladoo");

printf("\n RATE : %d",cur->rate);

}

if(cur->no==2)

{

printf("\n ITEM : Jillaebee");

printf("\n RATE : %d",cur->rate);

}

if(cur->no==3)

{

printf("\n ITEM : Milk sweet");

printf("\n RATE : %d",cur->rate);

}

prev=cur;

cur=cur->next;

}

cur=prev;

printf("\nReverse traversal:\n");

while(cur!=start)

{

if(cur->no==1)

{

printf("\n ITEM : Ladoo");

printf("\n RATE : %d",cur->rate);

}

if(cur->no==2)

{

printf("\n ITEM : Jillaebee");

printf("\n RATE : %d",cur->rate);

}

if(cur->no==3)

{

printf("\n ITEM : Milk sweet");

printf("\n RATE : %d",cur->rate);

}

cur=cur->prev;

}

if(cur->no==1)

{

printf("\n ITEM : Ladoo");

printf("\n RATE : %d",cur->rate);

}

if(cur->no==2)

{

printf("\n ITEM : Jillaebee");

printf("\n RATE : %d",cur->rate);

}

if(cur->no==3)

{

printf("\n ITEM : Milk sweet");

printf("\n RATE : %d",cur->rate);

}

}

OUTPUT:-





