Doubly Linked List

//DLL

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

void add\_beg();

void delete\_();

void display();

void add\_bet();

void del\_bet();

void add\_end();

void del\_end();

struct node

{

int data;

struct node \*flink;

struct node \*rlink;

}\*LIST=NULL,\*p,\*q;

void main()

{

int ch;

clrscr();

do

{

printf("1.Add\_Beg 2.Del\_Beg 3.Display 4.Exit 5.Add\_Bet 6.Del\_Bet 7.Add\_End 8.Del\_End\n");

printf("ENTER THE CHOICE:");

scanf("%d",&ch);

switch(ch)

{

case 1:

add\_beg();

break;

case 2:

delete\_();

break;

case 3:

display();

break;

case 5:

add\_bet();

break;

case 6:

del\_bet();

break;

case 7:

add\_end();

break;

case 8:

del\_end();

break;

}

}while(ch!=4);

getch();

}

void add\_beg()

{

int a;

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

printf("\nEnter Data : ");

scanf("%d",&a);

q->data=a;

q->rlink=NULL;

q->flink=LIST;

LIST->rlink=q;

LIST=q;

}

void delete\_()

{

if(LIST==NULL)

{

printf("\nUNDERFLOW\n");

}

else

{

q=LIST;

LIST=LIST->flink;

q->flink=NULL;

free(q);

}

}

void display()

{

p=LIST;

printf("Elements are :\n ");

if(LIST==NULL)

printf("LIST is empty\n");

else

{

while(p)

{

printf("%d-->",p->data);

p=p->flink;

}

printf("\n");

}

}

void add\_bet()

{

int pos,i,a;

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

printf("\nEnter Data : ");

scanf("%d",&a);

printf("\nEnter Position : ");

scanf("%d",&pos);

q->data=a;

p=LIST;

for(i=1;i<pos-1;i++)

{

p=p->flink;

}

q->flink=p->flink;

q->rlink=p;

p->flink=q;

p->flink->rlink=q;

}

void del\_bet()

{

int pos,i;

if(LIST==NULL)

{

printf("\nUNDERFLOW\n");

goto z;

}

printf("Enter Position : ");

scanf("%d",&pos);

p=LIST;

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

for(i=1;i<pos-1;i++)

{

p=p->flink;

}

q=p->flink;

p->flink=q->flink;

q->flink->rlink=p;

free(q);

z:

}

void add\_end()

{

int a;

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

if(LIST==NULL)

{

printf("\nUNDERFLOW\n");

goto z\_;

}

printf("\nEnter Data : ");

scanf("%d",&a);

q->data=a;

q->flink=NULL;

p=LIST;

while(p->flink!=NULL)

p=p->flink;

p->flink=q;

q->rlink=p;

z\_:

}

void del\_end()

{

if(LIST==NULL)

{

printf("\nUNDERFLOW\n");

goto y;

}

else if(LIST->flink==NULL)

{

q=LIST;

q->rlink->flink=NULL;

free(LIST);

goto y;

}

p=LIST;

while(p->flink!=NULL)

p=p->flink;

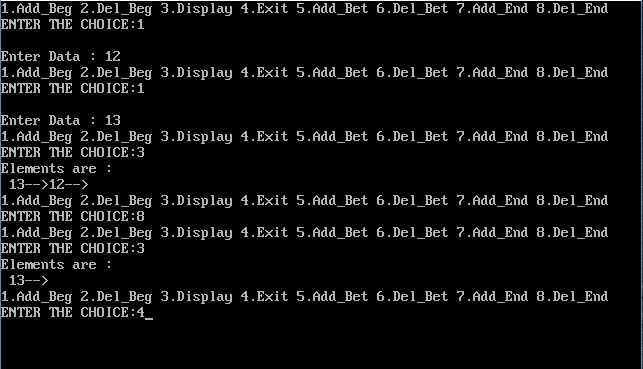
p->rlink->flink=NULL;

free(p);

y:

}

OUTPUT:



Circular Linked List

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

typedef struct Node

{

int info;

struct Node \*next;

}node;

node \*front=NULL,\*rear=NULL,\*temp;

void create();

void del();

void display();

void main()

{

int chc;

clrscr();

do

{

printf("\n1-Enqueue 2-Dequeue 3-Display 4-Exit\n");

printf("Enter your choice : ");

scanf("%d",&chc);

switch(chc)

{

case 1:

create();

break;

case 2:

del();

break;

case 3:

display();

break;

case 4:

return 1;

default:

printf("\nInvalid choice :");

}

}while(1);

}

void create()

{

node \*newnode;

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

printf("Enter Data : ");

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

newnode->next=NULL;

if(rear==NULL)

front=rear=newnode;

else

{

rear->next=newnode;

rear=newnode;

}

rear->next=front;

}

void del()

{

temp=front;

if(front==NULL)

printf("\nUnderflow :");

else

{

if(front==rear)

{

printf("\n%d Deleted\n",front->info);

front=rear=NULL;

}

else

{

printf("\n%d Deleted\n",front->info);

front=front->next;

rear->next=front;

}

temp->next=NULL;

free(temp);

}

}

void display()

{

temp=front;

if(front==NULL)

printf("\nEmpty");

else

{

printf("The Elements are : \n");

for(;temp!=rear;temp=temp->next)

printf("%d--> ",temp->info);

printf("%d--> ",temp->info);

}

}

OUTPUT:

