Circular Queue

//Circular Queue

#include<stdio.h>

#include<conio.h>

int front=-1,rear=-1,num;

int a[10];

void enqueue(int num1)

{

if(front==-1 )

{

front=rear=0;

a[front]=num1;

}

else

{

rear=(rear+1)%10;

a[rear]=num1;

}

}

void dequeue()

{

if(front==-1)

{

printf("Queue is Empty\n");

}

else

{

if(front==rear)

{

front=rear=-1;

}

else

front=(front+1)%10;

}

}

void display()

{

int i;

if(front==-1)

printf("Queue is empty\n");

else

{

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

if(rear<front)

{

for(i=front;i<=9;i++)

printf("%d\t",a[i]);

for(i=0;i<=rear;i++)

printf("%d\t",a[i]);

}

else

{

for(i=front;i<=rear;i++)

printf("%d\t",a[i]);

}

}

printf("\n");

}

void main()

{

int ch,num;

clrscr();

do

{

printf("1-Enqueue 2-Dequeue 3-Display 4-Exit\n");

printf("Enter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("Enter Element :");

scanf("%d",&num);

enqueue(num);

break;

case 2:

dequeue();

break;

case 3:

display();

break;

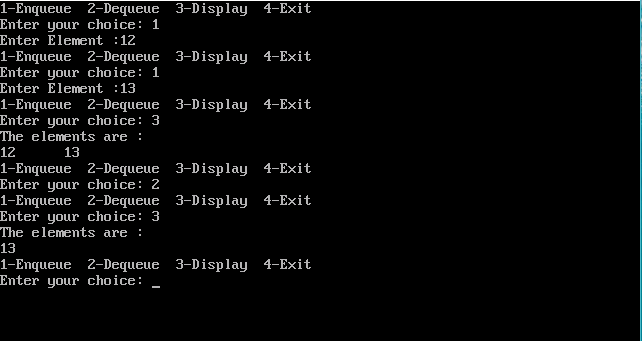
}

}while(ch!=4);

getch();

}

OUTPUT:



Queue with Linked List

#include<stdio.h>

#include<conio.h>

#include<malloc.h>

typedef struct node

{

int data;

struct node \*next;

}queue;

queue \*front=NULL,\*rear=NULL,\*newnode,\*temp;

void enqueue(int num1)

{

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

newnode->data=num1;

newnode->next=NULL;

if(front==NULL)

{

front=newnode;

rear=newnode;

}

else

{

rear->next=newnode;

rear=newnode;

}

}

void dequeue()

{

if(front==NULL)

{

printf("Queue is empty\n");

}

else

{

temp=front;

front=front->next;

free(temp);

}

}

void display()

{

if(front==NULL)

{

printf("Queue is empty\n");

}

else

{

temp=front;

printf("The elements are :\n-->");

while(temp!=NULL)

{

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

temp=temp->next;

}

Printf(“\n\n”);

}

}

void main()

{

clrscr();

printf(“Enqueue : 5 6 7 8\n”);

enqueue(5);

enqueue(6);

enqueue(7);

enqueue(8);

printf(“Call Display\n\n”);

display();

printf(“Dequeue\n\n”);

dequeue();

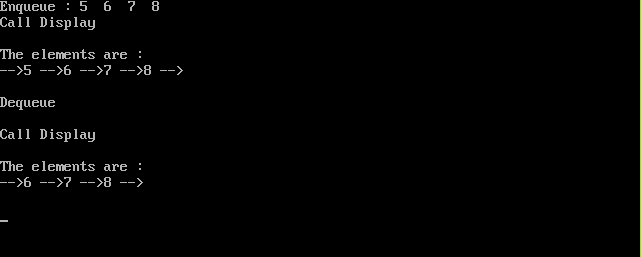
printf(“Call Display\n\n”);

display();

getch();

}

OUTPUT:



Double Ended Queue