**INPUT**

**Simple Queue implementation using Array and Linked List**  
#include<iostream>

#include<stdio.h>

using namespace std;

class Queue{

int front;

int rear;

int size;

int arr[20];

public:

Queue()

{

front=rear=-1; }

void enQueue(int data,int size);

void deQueue();

void display(); };

void Queue::enQueue(int data,int size)

{

this->size=size;

if(front==-1)

{

front=rear=0; }

if(rear==size)

{

cout<<"\n Full!"; }

if(size>rear)

{

arr[rear]=data;

++rear;

cout<<"\nInsetred Data"; } }

void Queue::deQueue()

{

if(front==-1 || front==size)

cout<<"\n No! Data";

else{

front++;

cout<<"\n Deleted!"; } }

void Queue::display(){

if(front!=rear)

{

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

{

cout<<arr[i]<<","; }

}else{

cout<<"\n No Data Exists!!!"; } }

int main()

{

int choice;

int size;

Queue obj;

int data;

cout<<"\n Enter Size of array";

cin>>size;

do{

cout<<"\n\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*";

cout<<"\n 1) Insert an Element";

cout<<"\n 2)Remove Element";

cout<<"\n 3)Display Data";

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n Enter your choice";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\n Enter Data to be Inserted:";

cin>>data;

obj.enQueue(data,size);

break;

case 2:

obj.deQueue();

break;

case 3:

obj.display();

break;

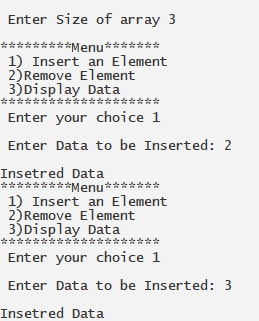
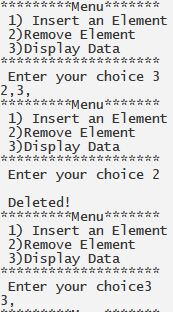
default:

cout<<"\n Plz enter proper value!"; } }

while(choice!=4);

return 0; }

**OUTPUT**

**INPUT**

#include<iostream>

using namespace std;

class Node{

public:

int data;

Node \*next;

Node() { }

Node(int data)

{

this->data=data;

next=NULL; } };

class Queue{

Node \*front;

Node \*rear;

public:

Queue()

{

front=NULL;

rear=NULL; }

void enQueue(int data);

void deQueue();

void display(); };

void Queue::enQueue(int data)

{

Node \*newRec=new Node(data);

if(front==NULL)

{

front=rear=newRec; }

else{

rear->next=newRec;

rear=newRec; }

cout<<"\n Inserted"; }

void Queue::deQueue()

{

Node \*delRec=front;

if(front==NULL)

{

cout<<"\n Underflow!!"; }else{

front=front->next;

cout<<"\n Deleted!!"; }

delete delRec; }

void Queue::display(){

if(front!=NULL)

{

for(Node \*i=front;i!=NULL;i=i->next)

{

cout<<"\t"<<i->data; } }

else{

cout<<"\n No data Exists"; } }

int main()

{

int choice;

Queue obj;

int data;

do{

cout<<"\n\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*";

cout<<"\n1) Insert an Element";

cout<<"\n2) Remove Element";

cout<<"\n3) Display Data";

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n Enter your choice:";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\n Enter Data to be Inserted:";

cin>>data;

obj.enQueue(data);

break;

case 2:

obj.deQueue();

break;

case 3:

obj.display();

break;

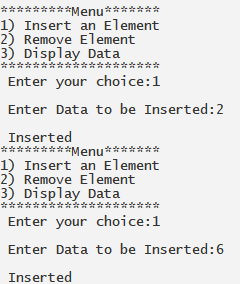
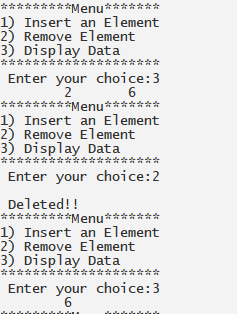
default:

cout<<"\n Plz enter proper value!"; } }

while(choice!=4);

return 0; }

**OUTPUT**

**INPUT**

**Circular Queue implementation using Array and Linked List**

#include<iostream>

#include<stdio.h>

using namespace std;

class Queue{

int front;

int rear;

int size;

int arr[20];

public:

Queue()

{

front=rear=-1; }

void enQueue(int data,int size);

void deQueue();

void display(); };

void Queue::enQueue(int data,int size)

{

this->size=size;

if(front==rear)

{

front=rear=0; }

if(rear==size)

{

cout<<"\n Full!"; }

if(size>rear)

{

arr[rear]=data;

++rear;

cout<<"\nInsetred Data"; } }

void Queue::deQueue()

{

if(front==-1 || front==size)

cout<<"\n No! Data";

else{

front++;

cout<<"\n Deleted!"; } }

void Queue::display(){

if(front!=rear)

{

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

{

cout<<arr[i]<<" "; } }else{

cout<<"\n No Data Exists!!!"; } }

int main()

{

int choice;

int size;

Queue obj;

int data;

cout<<"\n Enter Size of array";

cin>>size;

do{

cout<<"\n\*\*\*\*\*\*\*\*\*Menu\*\*\*\*\*\*\*";

cout<<"\n 1) Insert an Element";

cout<<"\n 2) Remove Element";

cout<<"\n 3) Display Data";

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n Enter your choice";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\n Enter Data to be Inserted:";

cin>>data;

obj.enQueue(data,size);

break;

case 2:

obj.deQueue();

break;

case 3:

obj.display();

break;

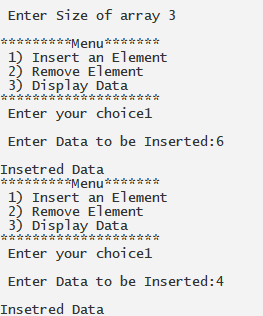
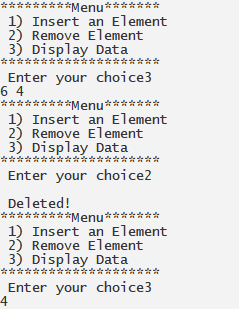
default:

cout<<"\n Plz enter proper value!"; } }

while(choice!=4);

return 0; }

**OUTPUT**

**INPUT**

#include<iostream>

#define SIZE 100

using namespace std;

class node

{

public:

node()

{

next = NULL; }

int data;

node \*next;

}\*front=NULL,\*rear=NULL,\*n,\*temp,\*temp1;

class cqueue

{

public:

void insertion();

void deletion();

void display(); };

void cqueue::insertion()

{

n=new node[sizeof(node)];

cout<<"\nEnter the Element: ";

cin>>n->data;

if(front==NULL)

{

front=n; }

else

{

rear->next=n; }

rear=n;

rear->next=front; }

void cqueue::deletion()

{

int x;

temp=front;

if(front==NULL)

{

cout<<"\nCircular Queue Empty!!!"; }

else

{

if(front==rear)

{

x=front->data;

delete(temp);

front=NULL;

rear=NULL; }

else

{

x=temp->data;

front=front->next;

rear->next=front;

delete(temp); }

cout<<"\nElement "<<x<<" is Deleted";

display(); } }

void cqueue::display()

{

temp=front;

temp1=NULL;

if(front==NULL)

{

cout<<"\n\nCircular Queue Empty!!!"; }

else

{

cout<<"\n\nCircular Queue Elements are:\n\n";

while(temp!=temp1)

{

cout<<temp->data<<" ";

temp=temp->next;

temp1=front; } } }

int main()

{

cqueue cqobj;

int ch;

do

{

cout<<"\n\n\tMain Menu";

cout<<"\n##########################";

cout<<"\n1. Insert\n2. Delete\n3. Display\n4. Exit\n\nEnter Your Choice: ";

cin>>ch;

switch(ch)

{

case 1:

cqobj.insertion();

cqobj.display();

break;

case 2:

cqobj.deletion();

break;

case 3:

cqobj.display();

break;

case 4:

break;

default:

cout<<"\n\nWrong Choice!!! Try Again."; } }

while(ch!=4);

return 0;

}

**OUTPUT**

