

Circular Queue Using Array-

Enqueue And Dequeue -

```
#include <iostream>
using namespace std;

class circularQueue{
public:
    const int n=100;
    int arr[100];
    int front;
    int rear;

    circularQueue(){
        front=-1;
        rear=-1;
    }

    bool isFull(){
        return(front==(rear+1)% n);
    }

    bool isEmpty(){
        return(front== -1);
    }

    void Enqueue(int data){
        if(isFull()){
            cout<<"Queue is full"<<endl;
        }
        if(isEmpty()){
            front=0;
        }
    }
};
```

```
}

rear=(rear+1)%n;
arr[rear]=data;
cout << "Inserted " << data << endl;

}

void Dequeue() {
    if (isEmpty()) {
        cout << "Queue is empty" << endl;
        return;
    }

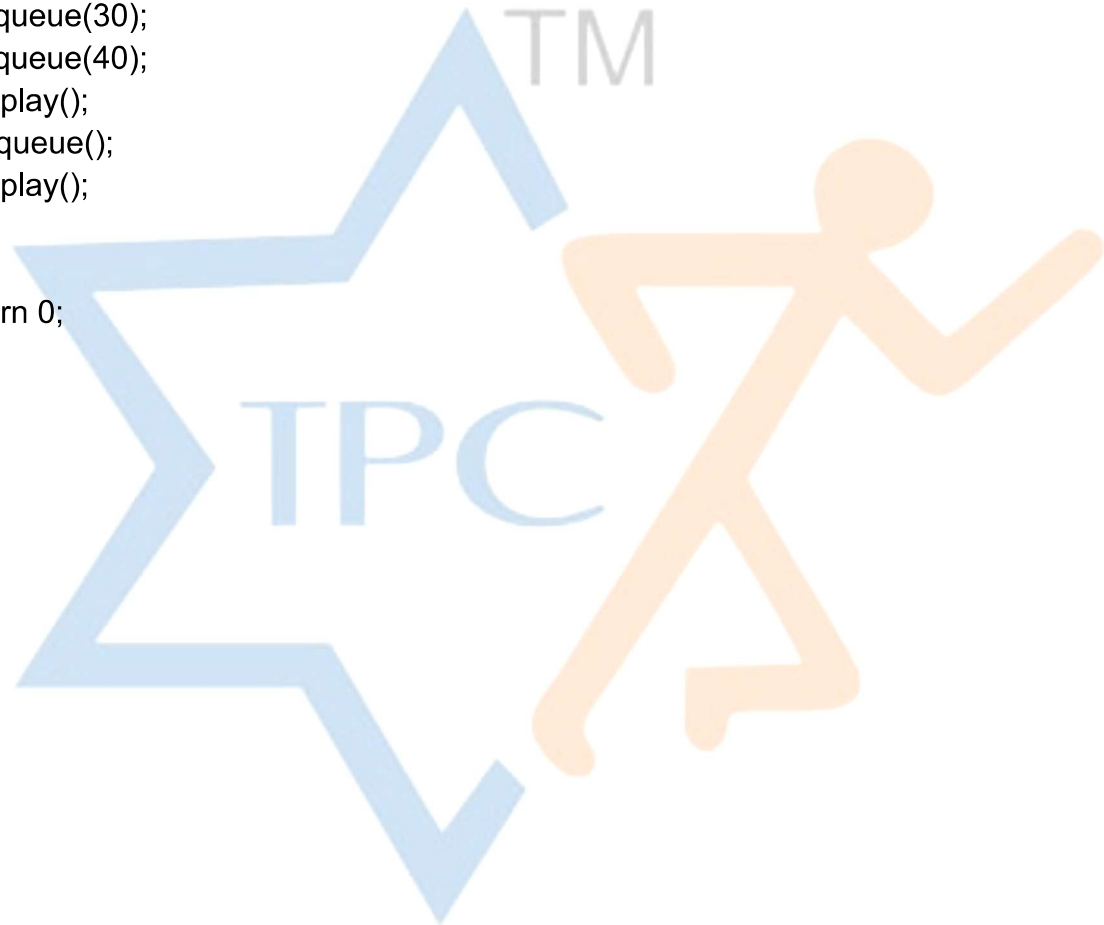
    cout << "Deleted element: " << arr[front] << endl;

    if (front == rear) {
        // Only one element was present
        front = rear = -1;
    } else {
        front = (front + 1) % n;
    }
}

void Display(){
    if(isEmpty()){
        cout<<"List is Empty"<<endl;
    }
    cout << "Queue elements: "<<endl;
    int i=front;
    while(true){
        cout<<arr[i]<<" ";
        if(i ==rear) break;
        i=(i + 1)%n;
    }
}
```

};

```
int main() {  
    circularQueue q;  
    q.Enqueue(20);  
    q.Enqueue(30);  
    q.Enqueue(40);  
    q.Display();  
    q.Dequeue();  
    q.Display();  
  
    return 0;  
}
```



www.tpcglobal.in