**SAIRA KOUSAR**

**BSCS -3**

**(55503)**

**DSA LAB # 11**

……………………………………………………………………………………………………………………………………………………….

**Task #01**

#include<iostream>

using namespace std;

const int size=6;

class queue{

public:

int list[size];

int rear;

int front;

public:

queue(){

rear = 0;

front = 0;

}

bool Empty(){

if(rear==front || rear == 0){

cout<<" list is empty "<<endl;

return true;

}

cout<<" list is not empty "<< endl;

return false;

}

bool full(){

if( rear=size-1){

cout<<" list is full"<<endl;

return true;

}

cout<<" list is not full"<<endl;

return false;

}

void enqueue(int val){

if(rear!=size)

list[rear]=val;

rear++;

cout<<" enqueue value is:"<<val<<endl;

}

void dequeue(){

if ( rear != 0 || rear!= front){

//cout<<" dequeue is :"<<list[front];

front++;

}

else{

cout<<" list is empty "<<endl;

}

}

void display(){

if ( rear != 0 || rear!= front){

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

cout<< list[i]<<endl;

}

}

}

};

int main(){

queue q;

cout<<" list of enqueue values :"<<endl;

q.enqueue(1);

q.enqueue(2);

q.enqueue(3);

q.enqueue(4);

q.enqueue(5);

q.full();

cout<<" values of dequeue :"<<endl;

q.dequeue();

q.dequeue();

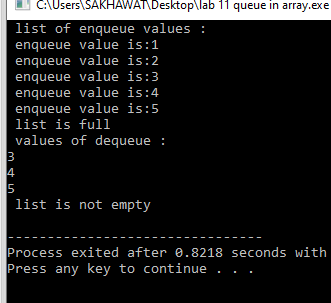
q.display();

q.Empty();

return 0;

}

**Output:**



**Task #02**

#include <iostream>

using namespace std;

class Queue {

struct Node {

int data;

Node\* next;

};

Node\* front;

Node\* rear;

public:

Queue() {

front = rear = NULL;

}

void enqueue(int data) {

Node\* newNode = new Node();

newNode->data = data;

newNode->next = NULL;

if (rear == NULL) {

front = rear = newNode;

return;

}

rear->next = newNode;

rear = newNode;

}

void dequeue() {

if (front == NULL) {

cout << "Queue is Empty" << endl;

return;

}

Node\* temp = front;

cout << "Dequeue: " << front->data << endl;

front = front->next;

if (front == NULL) {

rear = NULL;

}

delete temp;

}

void empty() {

if (front == NULL) {

cout << "Queue is Empty" << endl;

} else {

cout << "Queue is not Empty" << endl;

}

}

void display() {

if (front == NULL) {

cout << "Queue is Empty" << endl;

return;

}

Node\* temp = front;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

};

int main() {

Queue q;

cout<<" ....queue wih linklist... "<< endl;

cout<<" enqueue values are:"<< endl;

q.enqueue(10);

q.display();

q.enqueue(20);

q.display();

q.enqueue(30);

q.display();

cout<<" ======================="<< endl;

q.dequeue();

cout<<" list after dequeue :"<< endl;

q.display();

cout<<" ======================="<< endl;

q.empty();

return 0;

}

**Output**:

