#include <iostream>

using namespace std;

// Defining Size Of Queue

int const SIZE=5;

// Creating Class With Name Queue

class Queue

{

private:

int items[SIZE], front, rear;

public:

// Creatiing Constructor Of Class Queue

Queue()

{

front = -1; // Initializing front

rear = -1; // Initializing rear

}

// Check whether the queue is full or not

bool isFull()

{

if (front == 0 && rear == SIZE - 1) {

return true;

}

if (front == rear + 1)

{

return true;

}

return false;

}

// Check whether the queue is empty or not

bool isEmpty()

{

if (front == -1)

return true;

else

return false;

}

// Function For Inserting An Element

void enQueue(int element)

{

if (isFull())

{

cout << "Queue Is Full. You Cannot Insert Element.";

}

else

{

if (front == -1)

front = 0;

rear = (rear + 1) % SIZE;

items[rear] = element;

cout << endl;

cout<< "Inserted " << element << endl;

}

cout <<"\n";

}

// Function For Deleting An Element

int deQueue()

{

int element;

// Checking whether queue is empty or not

if (isEmpty())

{

cout << "Queue Is Empty. You Cannot Delete Element." << endl;

return (-1);

}

else

{

element = items[front];

if (front == rear)

{

front = -1;

rear = -1;

}

// Queue has only one element,

// so we reset the queue after deleting it.

else

{

front = (front + 1) % SIZE;

}

return (element);

}

cout<<"\n";

}

// Display Function

void display()

{

// Function to display status of Circular Queue

int i;

// Checking whether Queue is empty or not

if (isEmpty())

{

cout << endl

<< "Queue is Empty." << endl;

}

else

{

cout<< "Front -> " << front<<endl; // Displaying front

cout<< "Items -> ";

for (i = front; i != rear; i = (i + 1) % SIZE)

{

cout<<items[i]<<" ";

}

cout<<items[i]; // Displaying items in the Queue

cout<<"\nRear -> " << rear; // Displaying rear

}

cout<<"\n";

}

};

// Main Function

int main()

{

Queue q; // Creating object of class Queue

int choice, element;

do

{

cout<<"\n Which Operation You Want To Perform:";

cout<<" \n 1) Insertion (Enqueue).\n 2) Deletion (Dequeue). \n 3) Display. \n 4) Exit.";

cout<<"\n Enter Your Choice: ";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\nEnter element you want to insert: ";

cin>>element;

q.enQueue(element);

break;

case 2:

q.deQueue();

break;

case 3:

cout<<"\nDisplaying elements: \n";

q.display();

break;

case 4:

cout<<"\nYou Have Successfully Exitted.....";

break;

}

}

while (choice!=4);

return 0;

}