//circularqueue.h

class CCirularQueue

{

int front, rear;

int \*cqArray;

int MAX = 5;

public:

CCirularQueue();

CCirularQueue(int);

int insert(int);

int remove();

int display();

int isFull();

int isEmpty();

~CCirularQueue();

};

// circularqueue.cpp

#include "pch.h"

#include "circularueue.h"

#include <iostream>

using namespace std;

CCirularQueue::CCirularQueue()

{

front = -1;

rear = -1;

cqArray = new int[MAX];

}

//CCirularQueue::CCirularQueue(int MAX)

//{

// front = -1;

// rear = -1;

// this->MAX = MAX;

// cqArray = new int[MAX];

//}

int CCirularQueue::isFull()

{

if (((rear+1)%MAX == front) || (front == -1 && rear == (MAX-1)))

return 1;

else

return 0;

}

int CCirularQueue::isEmpty()

{

return front == rear;

}

int CCirularQueue::insert(int value)

{

if (isFull())

{

return 0;

}

else

{

rear = (rear + 1) % MAX;

cqArray[rear] = value;

return 1;

}

}

int CCirularQueue::remove()

{

if (isEmpty())

return 0;

else

{

front = (front + 1) % MAX;

cout << "Deleted value : "<< cqArray[front];

return cqArray[front];

}

}

int CCirularQueue::display()

{

for(int i=front+1;i<=rear;i++)

{

cout << cqArray[i] << " ";

}

return 0;

}

CCirularQueue::~CCirularQueue()

{

delete[] cqArray;

}

//cqmain.cpp

#include "pch.h"

#include "circularueue.h"

#include<conio.h>

#include <iostream>

using namespace std;

int main()

{

CCirularQueue cqObj;

int ch,value;

do

{

cout << "\n1.INSERT 2.DELETE 3.DISPLAY 4.EXIT \n";

cout << "Enter ch : ";

cin >> ch;

switch (ch)

{

case 1:

if (cqObj.isFull())

cout << "Queue is Full";

else

{

cout << "Enter value : ";

cin >> value;

cqObj.insert(value);

}

break;

case 2:

if (cqObj.isEmpty())

cout << "Queue empty......";

else

{

cqObj.remove();

}

break;

case 3:

if (cqObj.isEmpty())

cout << "Queue empty......";

else

{

cqObj.display();

}

break;

case 4:exit(0);

default:cout << "wrong input....";

}

} while (ch!=4);

\_getch();

return 0;

}