Lab Assignment 3

-----------------------------------------------------------------------------------------------------------------------------

Name: Prathamesh Sadashiv Gadekar

Class: SE-IT

Roll no.:14

Batch: S1

-----------------------------------------------------------------------------------------------------------------------------

PROBLEM STATEMENT:

Implement Circular Queue using Array. Perform following operations on it.

a) Insertion (Enqueue)

b) Deletion (Dequeue)

c) Display

-----------------------------------------------------------------------------------------------------------------------------

#include<iostream>

using namespace std;

class CircularQueue {

private:

int front;

int rear;

int arr[5];

int itemCount;

public:

CircularQueue()

{

itemCount = 0;

front = -1;

rear = -1;

for (int i = 0; i < 5; i++)

arr[i] = 0;

}

bool isEmpty();

bool isFull();

void enqueue(int val);

int dequeue();

int count();

void display();

};

int main()

{

CircularQueue q1;

int value, option;

do{

cout<<"\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n \n";

cout << "\t1. Enqueue()" << endl;

cout << "\t2. Dequeue()" << endl;

cout << "\t3. isEmpty()" << endl;

cout << "\t4. isFull()" << endl;

cout << "\t5. count()" << endl;

cout << "\t6. display()" << endl;

cout << "\n\nWhat operation do you want to perform?(Enter 0 for

exit) : " ;

cin >> option;

switch (option)

{

case 0:

break;

case 1:

cout <<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ENQUEUE

OERATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\nEnter an item to Enqueue in the Queue :

";

cin >> value;

q1.enqueue(value);

break;

case 2:

cout <<"\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEQUEUE

OERATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" ;

cout<<"\nDequeued Value : " << q1.dequeue() <<

endl;

break;

case 3:

if (q1.isEmpty())

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

else

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

break;

case 4:

if (q1.isFull())

cout << "Queue is Full" << endl;

else

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

break;

case 5:

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Count

Operation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\nCount of items in Queue : " <<

q1.count() << endl;

break;

case 6:

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Elements

Present in Circular Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n\n";

q1.display();

break;

default:

cout << "Enter Proper Option number " << endl;

}

} while (option != 0);

return 0;

}

//check queue is empty or not

bool CircularQueue::isEmpty()

{

if (front == -1 && rear == -1)

return true;

else

return false;

}

//check queue is full or not

bool CircularQueue::isFull()

{

if ((rear + 1) % 5 == front)

return true;

else

return false;

}

//fill queue from rear

void CircularQueue::enqueue(int val)

{

if (isFull())

{

cout << "Queue full" << endl;

return;

}

else if (isEmpty())

{

rear = 0;

front = 0;

arr[rear] = val;

}

else

{

rear = (rear + 1) % 5;

arr[rear] = val;

}

itemCount++;

}

//pop element from front

int CircularQueue::dequeue()

{

int x = 0;

if (isEmpty())

{

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

return x;

}

else if (rear == front)

{

x = arr[rear];

rear = -1;

front = -1;

itemCount--;

return x;

}

else

{

cout << "front value: " << front << endl;

x = arr[front];

arr[front] = 0;

front = (front + 1) % 5;

itemCount--;

return x;

}

}

//count total number of element in queue

int CircularQueue::count()

{

return (itemCount);

}

//display element in queue

void CircularQueue::display()

{

cout << "All values in the Queue are - " << endl;

for (int i = 0; i < 5; i++)

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

}

OUTPUT:

rmdstic@rmdstic-OptiPlex-3010:~$ g++ exp3\_53.cpp

rmdstic@rmdstic-OptiPlex-3010:~$ ./a.out

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ENQUEUE OERATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter an item to Enqueue in the Queue : 20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEQUEUE OERATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dequeued Value : 20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 3

Queue is Empty

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 4

Queue is not Full

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Count Operation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Count of items in Queue : 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Elements Present in Circular

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

All values in the Queue are -

0 0 0 0 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 0

OUTPUT:

rmdstic@rmdstic-OptiPlex-3010:~$ g++ exp3\_53.cpp

rmdstic@rmdstic-OptiPlex-3010:~$ ./a.out

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ENQUEUE OERATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter an item to Enqueue in the Queue : 20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DEQUEUE OERATION

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dequeued Value : 20

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 3

Queue is Empty

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 4

Queue is not Full

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Count Operation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Count of items in Queue : 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Elements Present in Circular

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

All values in the Queue are -

20 0 0 0 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Operations on

Queue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Enqueue()

2. Dequeue()

3. isEmpty()

4. isFull()

5. count()

6. display()

What operation do you want to perform?(Enter 0 for exit) : 0