**Assignment-3**

**Name:**

**Roll No:**

**Branch:** S.E. I.T.

**Date:**

**Title: Circular Queue**

/\*Implement Circular Queue using Array. Perform following operations on it.

a) Insertion (Enqueue)

b) Deletion (Dequeue)

c) Display (Note: Handle queue full condition by considering a fixed size of a queue).\*/

* **Program**

#include<iostream>

#define MAX 5

using namespace std;

class CQ\_array

{

int CQ[MAX], f, r;

public:

CQ\_array()

{

f = r = -1;

}

bool queueEmpty()

{

return(f == -1 && r == -1);

}

bool queueFull()

{

return(((r+1)%MAX) == f);

}

void enqueue(int data)

{

if(!queueFull())

{

if(f==-1) //1st element

f++;

r = (r+1)%MAX;

CQ[r] = data;

}

}

int dequeue()

{

int data = CQ[f];

f = (f+1)%MAX;

if(f == ((r+1)%MAX)) //last element is dequed

f = r = -1;

return data;

}

int queueFront()

{

return CQ[f];

}

int queueRear()

{

return CQ[r];

}

void Display()

{

if (queueEmpty())

cout<<"Queue is empty\n";

else

{

cout << "Front = " << f << ", Rear = " << r << endl;

cout<<"Queue elements are : ";

/\*if(r >= f)

for (int i = f; i <= r; i++)

cout << CQ[i] << " ";

else

{\*/

for (int i = f; i != r; i=(i+1)%MAX)

cout << CQ[i] << " ";

cout << CQ[r] << "\n";

//}

}

}

};

int main()

{

int ch, data;

CQ\_array q;

do

{

cout << "\n\nOperations on Circular Queue (Array)\n";

cout << "1) Enqueue (Insert an element) \n";

cout << "2) Dequeue (Delete an element) \n";

cout << "3) QueueFront (1st Element in Queue) \n";

cout << "4) QueueRear (Last Element in Queue) \n";

cout << "5) Queue Full \n";

cout << "6) Queue Empty \n";

cout << "7) Display elements in the queue \n";

cout << "8) Exit \n";

cout << "Enter your choice : ";

cin >> ch;

switch (ch)

{

case 1: if(!q.queueFull())

{

cout << "Enter data to insert in the Queue: ";

cin >> data;

q.enqueue(data);

}

else

cout << "Queue is FULL!! \n";

break;

case 2: if(q.queueEmpty())

cout << "Queue is EMPTY!! \n";

else

{

data = q.dequeue();

cout << "Dequeued data is: " << data << endl;

}

break;

case 3: if(!q.queueEmpty())

{

data = q.queueFront();

cout << "1st Element in the Queue is: " << data << endl;

}

else

cout << "Queue is EMPTY!! \n";

break;

case 4: if(!q.queueEmpty())

{

data = q.queueRear();

cout << "Last Element in the Queue is: " << data << endl;

}

else

cout << "Queue is EMPTY!! \n";

break;

case 5: if(q.queueFull())

cout << "Yes, Queue is FULL!! \n";

else

cout << "No, Queue is NOT FULL!! \n";

break;

case 6: if(q.queueEmpty())

cout << "Yes, Queue is EMPTY!! \n";

else

cout << "No, Queue is NOT EMPTY!! \n";

break;

case 7: q.Display();

break;

case 8: cout << "Exit" << endl;

break;

default: cout << "Invalid choice \n";

}

} while(ch!=8);

return 0;

}

***OUTPUT***

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 2

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 3

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 4

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 5

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 7

Front = 0, Rear = 3

Queue elements are : 2 3 4 5

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 2

Dequeued data is: 2

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 7

Front = 1, Rear = 3

Queue elements are : 3 4 5

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 3

1st Element in the Queue is: 3

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 4

Last Element in the Queue is: 5

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 5

No, Queue is NOT FULL!!

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 6

No, Queue is NOT EMPTY!!

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 2

Dequeued data is: 3

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 2

Dequeued data is: 4

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 2

Dequeued data is: 5

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 6

Yes, Queue is EMPTY!!

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 7

Queue is empty

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 3

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 4

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 5

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 6

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 1

Enter data to insert in the Queue: 7

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 5

Yes, Queue is FULL!!

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 7

Front = 0, Rear = 4

Queue elements are : 3 4 5 6 7

Operations on Circular Queue (Array)

1) Enqueue (Insert an element)

2) Dequeue (Delete an element)

3) QueueFront (1st Element in Queue)

4) QueueRear (Last Element in Queue)

5) Queue Full

6) Queue Empty

7) Display elements in the queue

8) Exit

Enter your choice : 8

Exit

=== Code Execution Successful ===