**Task # 1:** Design & implement all methods of Simple Queue.

**Solution**

using System;

namespace LAb8\_Queue

{

internal class queue

{

int[] a = new int[5];

int front = 0;

int rear = -1;

internal bool isempty()

{

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

{

return true;

}

else

{

return false;

}

}

internal bool isfull()

{

if (rear == a.Length - 1)

{

Console.WriteLine("Queue Is full");

return true;

}

else

{

return false;

}

}

internal int Enqueue(int data)

{

if (isfull())

{

return 0;

}

rear = rear + 1;

a[rear] = data;

Console.WriteLine("{0} Inserted ", data);

return 1;

}

internal void peak()

{

if (!isempty())

{

Console.WriteLine("Queue Peak Element:{0}", a[front]);

}

else

{

Console.WriteLine("Queue Is Empty");

}

}

internal int dequeue()

{

if (isempty())

{

return 0;

}

int data = a[front];

front = front + 1;

Console.WriteLine("Dequeue deleted Element:{0}", data);

return 1;

}

internal void count()

{

int count = 0;

if (!isempty())

{

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

{

count++;

}

Console.WriteLine("Total Element In Queue Are {0}", count);

}

else

{

Console.WriteLine("Queue Is Empty");

}

}

internal void Display()

{

if (!isempty())

{

Console.WriteLine("Total Element In Queue Are ");

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

{

Console.WriteLine($"{a[i]}");

}

}

else

{

Console.WriteLine("Queue Is Empty");

}

}

static void Main(string[] args)

{

queue q = new queue();

int res;

do

{

Console.WriteLine("Please Choose From Below");

Console.WriteLine(" 1 ) Enqueue");

Console.WriteLine(" 2 ) Dequeue");

Console.WriteLine(" 3 ) Isfull method");

Console.WriteLine(" 4 ) Isempty method");

Console.WriteLine(" 5 ) peak");

Console.WriteLine(" 6 ) count");

Console.WriteLine(" 7 ) Display");

Console.Write("Enter : ");

res = int.Parse(Console.ReadLine());

if (res == 1)

{

Console.Write("Please Enter Data : ");

q.Enqueue(Convert.ToInt32(Console.ReadLine()));

}

else if (res == 2)

{

q.dequeue();

}

else if (res == 3)

{

if (q.isfull())

{

Console.WriteLine("Queue Is Full");

}

}

else if (res == 4)

{

if (q.isempty())

{

Console.WriteLine("Queue Is Empty");

}

}

else if (res == 5)

{

q.peak();

}

else if (res == 6)

{

q.count();

}

else if (res == 7)

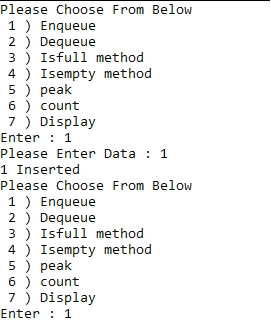
{

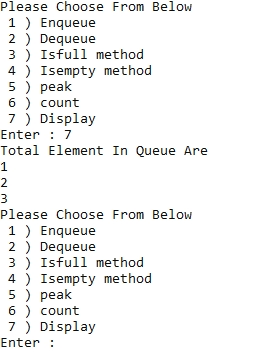
q.Display();

}

} while (res != 1 || res != 2 || res != 3 || res != 4 || res != 5 || res != 6 || res != 7);}}}

**Output**





**Task # 2** Write a code to implement messaging system that is capable to handle message requests, message capacity must be “5” and after that store messages in buffer. Methods to implement:

* Enqueue
* Dequeue
* Print Queue

**Solution**

using System;

using System.Collections;

namespace LAb8\_Queue

{

internal class queue

{

static int max = 5;

int[] per = new int[max];

public ArrayList buffer = new ArrayList();

int front = -1;

int rear = -1;

public bool isempty()

{

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

{

return true;

}

return false;

}

public bool isfull()

{

if (rear == per.Length - 1)

{

return true;

}

return false;

}

public void enqueue(int data)

{

if (!isfull())

{

if (isempty())

{

front = 0;

rear = 0;

per[rear] = data;

}

else

{

rear = rear + 1;

per[rear] = data;

}

}

else

{

Console.WriteLine("Queue Is Full ");

buffer.Add(data);

}

}

public void dequeue()

{

if (!isempty())

{

if (front == rear)

{

front = rear = -1;

}

else

{

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

{

per[i] = per[i + 1];

}

rear = rear - 1;

if (buffer.Count != 0)

{

rear = rear + 1;

per[4] = Convert.ToInt32(buffer[0]);

buffer.RemoveAt(0);

}

}

}

else

{

Console.WriteLine("Queue Is Empty!!! ");

Console.WriteLine();

}

}

public void print()

{

int count1 = 0;

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

{

Console.WriteLine(per[i]);

}

}

public void printarrlist()

{

for (int i = 0; i < buffer.Count; i++)

{

Console.WriteLine(buffer[i]);

}

}

static void Main(string[] args)

{

queue queue = new queue();

int res;

do

{

Console.WriteLine("---------------------------------------------------------------------------------------------");

Console.WriteLine(" 1) Enque");

Console.WriteLine(" 2) Deqeue");

Console.WriteLine(" 3) Print Queue");

Console.WriteLine(" 4) Print ArrayList");

Console.WriteLine("\n\n---------------------------------------------------------------------------------------------");

Console.Write("Enter : ");

res = Convert.ToInt32(Console.ReadLine());

if (res == 1)

{

Console.Write("Enter Your Data : ");

queue.enqueue(Convert.ToInt32(Console.ReadLine()));

}

else if (res == 2)

{

queue.dequeue();

Console.WriteLine("DeQueue operation Is Exeuated ");

}

else if (res == 3)

{

Console.WriteLine("---------------------------------------------------------------------------------------");

queue.print();

}

else if (res == 4)

{

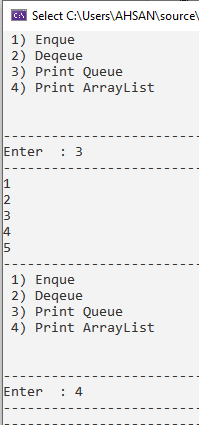
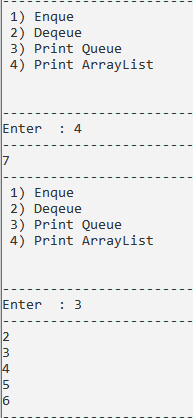
Console.WriteLine("---------------------------------------------------------------------------------------");

queue.printarrlist();

}

} while (res != 1 || res != 2 || res != 3 || res != 4);}}}

**Output**

****