Bahria University,

Karachi Campus

A picture containing text, room

Description automatically generated

LAB EXPERIMENT NO.

**08**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
|  | Queue |
| 1 | Design & implement all methods of Simple Queue |
| 2 | Design & implement all methods of Circular Queue. |
| 3 | Design and implement for Priority Queue.  Method 1: Ordering in/ after Enqueue method  Method 2: Separate queues for different priorities |

Submitted On:

**12 - 12 - 2022**

(Date: DD/MM/YY)

**TASK NO 1:** Design & implement all methods of Simple Queue

**SOLUTION:**

class Queue {

int[] a = new int[10];

int Front = -1;

int rear = -1;

public bool IsEmpty() {

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

{

return true;

}

else

{

return false;

}

}

public bool IsFull()

{

if (rear==a.Length-1)

{

return true;

}

else

{

return false;

}

}

public void enque(int data) {

if (IsFull()==true)

{

return;

}

else if (IsEmpty()==true)

{

rear = 0;

Front = 0;

a[rear] = data;

}

else

{

rear = rear + 1;

a[rear] = data;

}

}

public void Dequeue() {

if (IsEmpty()==true)

{

return;

}

else if (Front==rear)

{

Front = rear = -1;

}

else

{

Front = Front + 1;

}

}

public void printqueue() {

Console.WriteLine("Elements In Queues :");

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

{

Console.WriteLine(a[i] + " ");

}

} }

class Program

{

static void Main(string[] args)

{

Queue q = new Queue();

char ch,br;

do

{

Console.WriteLine("Chose Any Operation :\n1)Enqueue\n2)Dequeue\n3)Isempty\n4)Isfull\n5)Elements in Queue ");

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

if (option == 1)

{

do

{

Console.WriteLine("Enter Any number :");

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

q.enque(n);

Console.WriteLine("If You want to again[y/n]");

ch = char.Parse(Console.ReadLine());

} while (ch=='y'||ch=='Y');

}

else if (option==2)

{

q.Dequeue();

Console.WriteLine("After Dequeue Element:");

q.printqueue();

}

else if (option==3)

{

q.IsEmpty();

}

else if (option == 4)

{

q.IsFull();

}

else if (option == 5)

{

q.printqueue();

}

Console.WriteLine("Again operation perform :[T/n]");

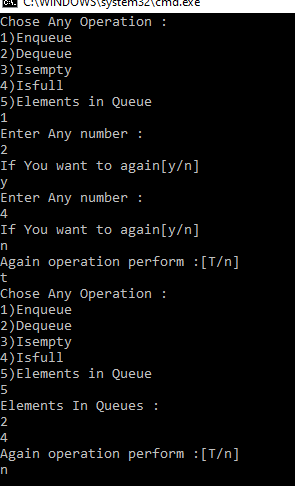
br = char.Parse(Console.ReadLine());

} while (br=='t'||br=='T');

Console.ReadLine();

}}}

**OUTPUT:**



**TASK NO 2:** Design & implement all methods of Circular Queue.

**SOLUTION:**

int[] a = new int[10];

int Front = -1;

int rear = -1;

public bool IsEmpty()

{

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

{

return true;

}

else

{

return false;

}

}

public bool IsFull()

{

if (Front == (rear + 1) % a.Length)

{

Console.WriteLine("Queue Is full");

return true;

}

else

{

return false;

}

}

public void enque(int data)

{

if (IsFull())

{

return;

}

else if (IsEmpty())

{

rear = 0;

Front = 0;

a[rear] = data;

}

else

{

rear = rear + 1 % a.Length;

a[rear] = data;

}

}

public void Dequeue()

{

if (IsEmpty())

{

return;

}

else if (Front == rear)

{

Front = rear = -1;

}

else

{

Front = Front + 1 % a.Length;

}

}

public void printqueue()

{

Console.WriteLine("Elements In Queues :");

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

{

Console.WriteLine(a[i] + " ");

}

}

public void count()

{

int count = 0;

if (IsEmpty())

{

Console.Write("Queue is Empty");

return;

}

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

{

count++;

}

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

}

public void peak()

{

Console.WriteLine($"Peak Element Is {a[Front + 1] % a.Length}");

}

}

class Program

{

static void Main(string[] args)

{

Queue q = new Queue();

char ch, br;

do

{

Console.WriteLine("Chose Any Operation :\n1)Enqueue\n2)Dequeue\n3)Isempty\n4)Isfull\n5)Elements in Queue\n6)Peek\n7)count ");

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

if (option == 1)

{

do

{

Console.WriteLine("Enter Any number :");

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

q.enque(n);

Console.WriteLine("If You want to again[y/n]");

ch = char.Parse(Console.ReadLine());

} while (ch == 'y' || ch == 'Y');

}

else if (option == 2)

{

q.Dequeue();

Console.WriteLine("After Dequeue Element:");

q.printqueue();

}

else if (option == 3)

{

if (q.IsEmpty())

{

Console.WriteLine("Queue is empty");

}

}

else if (option == 4)

{

if (q.IsFull())

{

Console.WriteLine("Queue is full");

}

}

else if (option == 5)

{

q.printqueue();

}

else if (option == 6)

{

q.peak();

}

else if (option == 7)

{

q.count();

}

Console.WriteLine("Again operation perform :[T/n]");

br = char.Parse(Console.ReadLine());

} while (br == 't' || br == 'T');

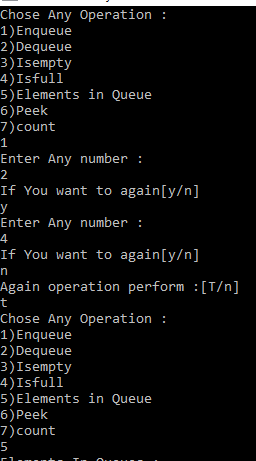
Console.ReadLine();

}

}

}

**OUTPUT:**

****

Text

Description automatically generated