//IMPLEMENTATION OF CIRCULAR QUEUE USING ARRAY //

ALGORITHM:

ENQUEUE:

Step 1: if(rear+1)%N front

Write overflow end if

Step 2: if front ==-1 &rear==-1

Set front =rear=0

Else if rear =max -1 and front!=0

Set rear =0

Set rear=(rear+1)%N

End if

Step 3: set queue[rear]=val

Step 4 : exit

DEQUEUE :

Step 1: if front =-1

Write under flow

End if

Step 2:set val=queue[front]

Step 3:if front=rear

Set front =rear-1

Else

If front ,v =-1

Set front =0

Else

Set front =front+1

End if

End if

Step :4 exit

PROGRAM:

#include<stdio.h>

#define SIZE 5

Int items[SIZE];

INT front=-1,rear=-1;

//check if the queue is full

Int is full()

{

If((front==rear+1)||(front==0&&rear==size-1))

Return 1;

Return 0;

}

//check if the queue is empty

Int isempty()

{

If(front==-1)

Return 1;

Return 0;

}

//adding an element

Void enqueue(int element)

{

If(isfull())

Printf(“\n queue is full!!\n”);

Else

{

If(front==-1)front =0;

Rear=(rear+1)%SIZE;

Items[rear]=element;

Printf(“\n inserted->%d,element);

}

}

//removing an element

Int dequeue()

{

Int element;

If(isempty())

{

Printf(“\n queue is empty !!\n”);

Return (-1);

}

Else

{

Element=items[front];

If(front==rear)

{

Front=-1;

Rear=-1;

}

//q has only one element

// queue after dequeue

Else

{

Front=(front+1) %SIZE

}

Printf(“\n deleted element->%d\n”,element);

Return(element);

}

}

// Display the queue

Void display()

{

Int I;

If(isempty())

Printf(“\n empty queue\n”);

Else

{

Printf(“\n front->d”,front);

Printf(“\n items->”);

For(!=front;i!=rear,i=(i+1)%SIZE)

{

Printf(“%d”,items[i]);

}

Printf(“d”,items[i]);

Printf(“\n Rear->%d\n”,rear);

}

}

Int main()

Dequeue();

Enqueue(1);

Enqueue(2);

Enqueue(3);

Enqueue(4);

Enqueue(5);

//fails to enqueue because front=0

&& rear==SIZE-1

Enqueue(6);’

Display();

Dequeue();

Display();

Enqueue(7);

Display();

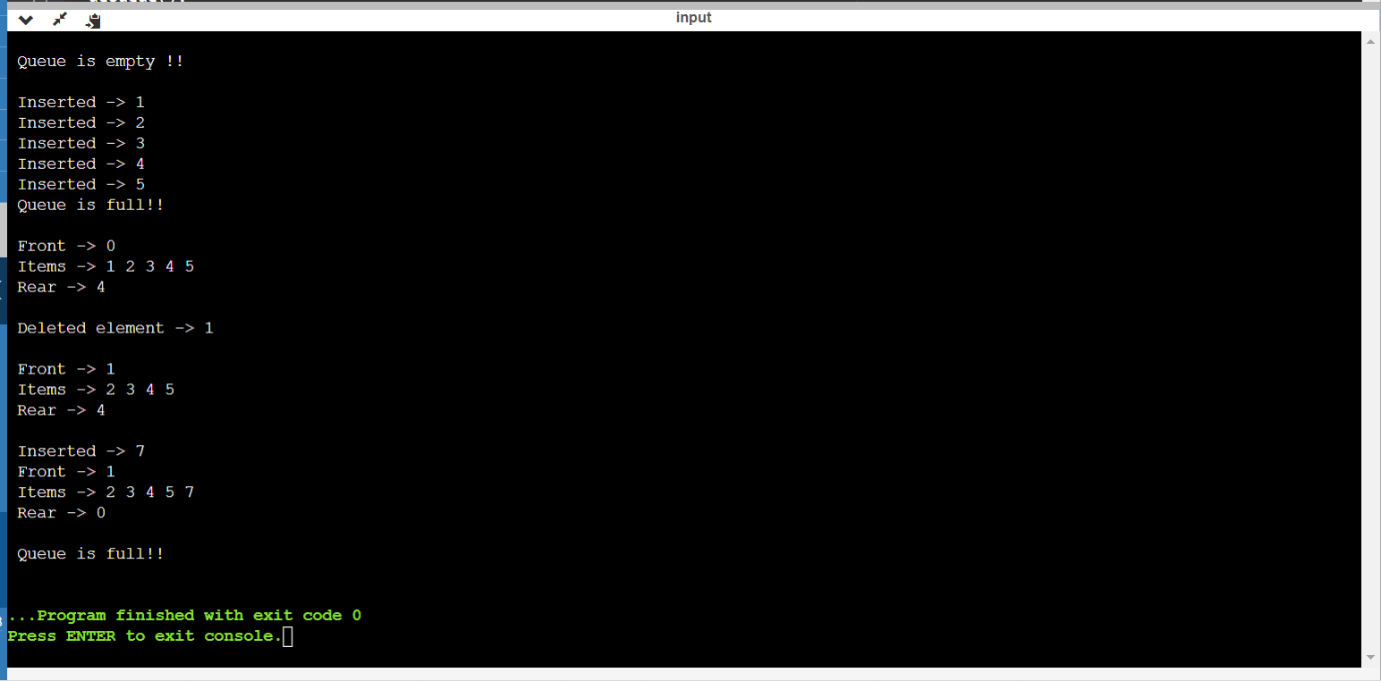
//fail to enqueue because front==rear+1

Enqueue(8);

Return 0;

}

OUTPUT:



GITHUB LINK: