11/01/2024

3 (B).Circular queue implementation

#include <stdio.h>

int q[50],front=-1,rear=-1,size;

void display();

void enqueue();

void dequeue();

void main()

{

int ch;

printf("Enter no. of elements:");

scanf("%d",&size);

while(ch!=4)

{

printf("1.Insert 2.Delete 3.Display 4.Exit\n");

printf("Enter your choice:");

scanf("%d",&ch);

switch (ch)

{

case 1:

enqueue();

break;

case 2:

dequeue();

break;

case 3:

display();

break;

}

}

printf("Exited");

}

void enqueue()

{

int item;

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

printf("Queue is full\n");

else

{

if(front == -1)

front=0;

printf("Enter element:");

scanf("%d",&item);

rear=(rear + 1)%size;

q[rear] = item;

}

}

void dequeue()

{

int ele;

if(front==-1)

printf("Queue is empty\n");

else

{

ele = q[front];

if(front==rear)

{

front = -1;

rear = -1;

}

else

front = (front+1) % size;

printf("Deleted element = %d\n",ele);

}

}

void display()

{

int i;

if(front == -1)

printf("Queue is empty");

else

{

printf("Front = %d\t",front);

printf("Rear = %d\n",rear);

printf("Queue:");

for(i=front;i!=rear;i=(i+1)%size)

printf("%d",q[i]);

printf("%d\n",q[i]);

}

}

OUTPUT:

