Circular Queue using array

VAISHAK S(R3-258)

**ALGORITHM**

**Input**: Elements of queue

**Output**: Insertion, deletion and displaying Queue

**Steps:**

MAX=10

queue[MAX]

front = -1

rear = -1

**insert(int ele)**

1. Start
2. if((front==0 && rear==MAX-1) or (front==rear+1))
3. print "Queue Overflow"
4. return
5. if(front==-1)
6. front = 0
7. rear = 0
8. else
9. if(rear==MAX-1)
10. rear=0
11. else
12. rear=rear+1
13. queue[rear]=ele
14. Stop

**deletion( )**

1. Start
2. if(front==-1)
3. print "Queue Underflow"
4. return
5. Print queue[front]
6. if(front==rear)
7. front=-1
8. rear=-1
9. else
10. if(front==MAX-1)
11. front=0
12. else
13. front=front+1
14. Stop

**display( )**

1. Start
2. Set front\_pos=front, rear\_pos=rear
3. if(front==-1)
4. print "Queue is empty"
5. return
6. Print "Queue elements: "
7. if(front\_pos<=rear\_pos)
8. while(front\_pos<=rear\_pos)
9. print queue[front\_pos]
10. front\_pos++
11. else
12. while(front\_pos<= MAX-1)
13. Print queue[front\_pos]
14. front\_pos++
15. front\_pos = 0
16. while(front\_pos<=rear\_pos)
17. Print queue[front\_pos]
18. front\_pos++
19. Stop

**main( )**

1. Start
2. Declare ch, ele
3. do

{

1. print "1.Insert

2.Delete

3.Display

4.Exit

Enter your choice : "

1. Read ch
2. If(ch==1)

insert(ele)

1. else if(ch==2)

deletion( )

1. Else if(ch==3)

display( )

1. else

print "Wrong choice"

1. } closing do loop

while(ch!=4)

1. Stop

**PROGRAM**

#include<stdio.h>

# define MAX 10

int queue[MAX];

int front = -1;

int rear = -1;

void insert(int ele)

{

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

{

printf("Queue Overflow\n");

return;

}

if(front==-1)

{

front = 0;

rear = 0;

}

else

{

if(rear==MAX-1)

rear=0;

else

rear=rear+1;

}

queue[rear]=ele;

}

void deletion()

{

if(front==-1)

{

printf("Queue Underflow\n");

return ;

}

printf("Element deleted from queue is : %d\n",queue[front]);

if(front==rear)

{

front=-1;

rear=-1;

}

else

{

if(front==MAX-1)

front=0;

else

front=front+1;

}

}

void display()

{

int front\_pos=front,rear\_pos=rear;

if(front==-1)

{

printf("Queue is empty\n");

return;

}

printf("Queue elements: ");

if(front\_pos<=rear\_pos)

while(front\_pos<=rear\_pos)

{

printf("%d ",queue[front\_pos]);

front\_pos++;

}

else

{

while(front\_pos<= MAX-1)

{

printf("%d ",queue[front\_pos]);

front\_pos++;

}

front\_pos = 0;

while(front\_pos<=rear\_pos)

{

printf("%d ",queue[front\_pos]);

front\_pos++;

}

}

printf("\n");

}

void main()

{

int ch,ele;

do

{

printf("1.Insert\n");

printf("2.Delete\n");

printf("3.Display\n");

printf("4.Quit\n");

printf("Enter your ch : ");

scanf("%d",&ch);

switch(ch)

{

case 1 :

printf("Enter the element: ");

scanf("%d", &ele);

insert(ele);

break;

case 2 :

deletion();

break;

case 3:

display();

break;

case 4:

break;

default:

printf("Wrong chn");

}

}while(ch!=4);

}