program of circular queue implementation

#include <stdio.h>

# define max 6

int queue[max];

int front=-1;

int rear=-1;

void enqueue(int element)

{

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

{

front=0;

rear=0;

queue[rear]=element;

}

else if((rear+1)%max==front)

{

printf("Queue is overflow..");

}

else

{

rear=(rear+1)%max;

queue[rear]=element;

}

}

int dequeue()

{

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

{

printf("\nQueue is underflow..");

}

else if(front==rear)

{

printf("\nThe dequeued element is %d", queue[front]);

front=-1;

rear=-1;

}

else

{

printf("\nThe dequeued element is %d", queue[front]);

front=(front+1)%max;

}

}

void display()

{

int i=front;

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

{

printf("\n Queue is empty..");

}

else

{

printf("\nElements in a Queue are :");

while(i<=rear)

{

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

i=(i+1)%max;

}

}

}

int main()

{

int choice=1,x;

while(choice<4 && choice!=0)

{

printf("\n Press 1: Insert an element");

printf("\nPress 2: Delete an element");

printf("\nPress 3: Display the element");

printf("\nEnter your choice");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter the element which is to be inserted");

scanf("%d", &x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

break;

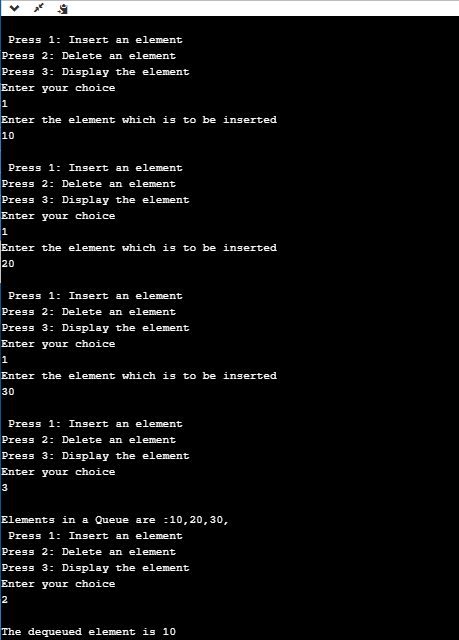
}

}

return 0;

}

output pf program



**Algorithm to insert an element in a circular queue**

**Step 1:** IF (REAR+1)%MAX = FRONT  
Write " OVERFLOW "  
Goto step 4  
[End OF IF]

**Step 2:** IF FRONT = -1 and REAR = -1  
SET FRONT = REAR = 0  
ELSE IF REAR = MAX - 1 and FRONT ! = 0  
SET REAR = 0  
ELSE  
SET REAR = (REAR + 1) % MAX  
[END OF IF]

**Step 3:** SET QUEUE[REAR] = VAL

**Step 4:** EXIT

**Algorithm to delete an element from the circular queue**

**Step 1:** IF FRONT = -1  
Write " UNDERFLOW "  
Goto Step 4  
[END of IF]

**Step 2:** SET VAL = QUEUE[FRONT]

**Step 3:** IF FRONT = REAR  
SET FRONT = REAR = -1  
ELSE  
IF FRONT = MAX -1  
SET FRONT = 0  
ELSE  
SET FRONT = FRONT + 1  
[END of IF]  
[END OF IF]

**Step 4:** EXIT