**WEEK 04**

**WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display**

**The program should print appropriate messages for queue empty and queue overflow conditions**

#include <stdio.h>

#include <stdlib.h>

#define size 5

int f=-1;

int r=-1;

int q[size];

void enqueue(int item)

{

    if(f==(r+1)%size)

    printf("Queue is full");

    else{

        r=(r+1)%size;

        q[r]=item;

        if(f==-1)

            f=f+1;

    }

}

void dequeue()

{

    if(f==-1)

    printf("Queue is empty");

    else{

        printf("\n Elements deleted is %d ",q[f]);

        if(f==r)

        {

            f=-1;

            r=-1;

        }

        else{

            f=(f+1)%size;

        }

    }

}

void display()

{

    int i;

    if(f==-1)

        printf("Queue is empty");

    else{

        printf("\n Content of queue : \n");

        for(i=f ; i!=r ; i=(i+1)%size)

        printf("%d \t",q[i]);

        printf("%d \t", q[r]);

    }

}

int main()

{

    int ch , item;

    for(;;)

    {

        printf("\n 1. Insert");

        printf("\n 2. Delete");

        printf("\n 3. Display");

        printf("\n 4. Exit");

        printf("\n Read choice : ");

        scanf("%d",&ch);

        switch(ch)

        {

            case 1 : printf("\n Enter element to be inserted : ");

                    scanf("%d",&item);

                    enqueue(item);

                    break;

            case 2 : dequeue();

                    break;

            case 3 : display();

                    break;

            default : exit(0);

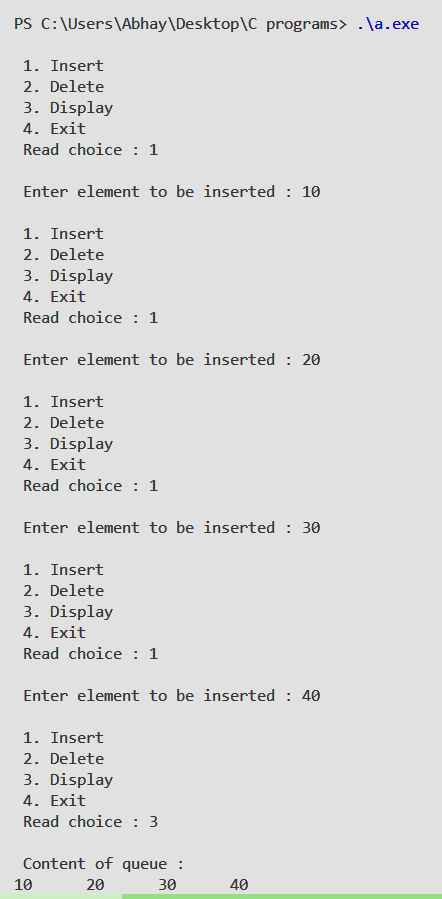
        }

    }

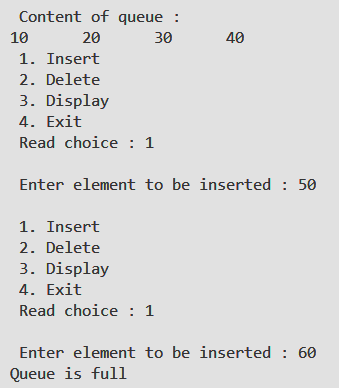
    return 0;

}

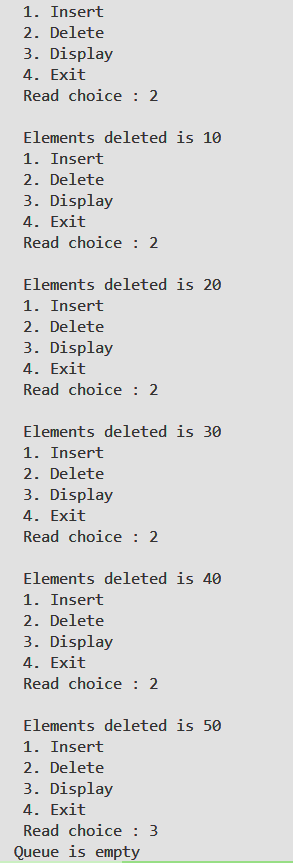
**Enqueue into the circular queue and displaying**



**Circular queue overflow**



**Deleting elements form the queue till queue is empty**



**Checking the Circular list working**

**First we enter 3 elements and then delete once and enqueue 40 to the queue .**

