**WEEK 03**

**WAP to simulate the working of a 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.**

**Code:**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 5

int front = -1 , rear = -1 ;

int q[SIZE];

void enqueue (int item)

{

if(rear==SIZE-1)

printf("\n Queue is full ");

else{

rear=rear+1;

q[rear]=item;

if(front==-1)

front=front+1;

}

}

void dequeue()

{

int del;

if(front==-1)

printf("\n Queue is empty");

else{

del=q[front];

printf("\n Element deleted is : %d",del);

if(front==rear)

{

front=-1;rear=-1;

}

else{

front=front+1;

}

}

}

void display()

{

int i;

if(front==-1)

printf("\n Queue is empty");

else{

printf("\n Queue content \n");

for (i=front;i<=rear;i++)

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

}

}

int main()

{

int item,ch;

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 Read element to be inserted ");

scanf("%d", & item);

enqueue(item);

break;

case 2 : dequeue();

break;

case 3 : display();

break;

default : exit (0);

}

}

return 0;

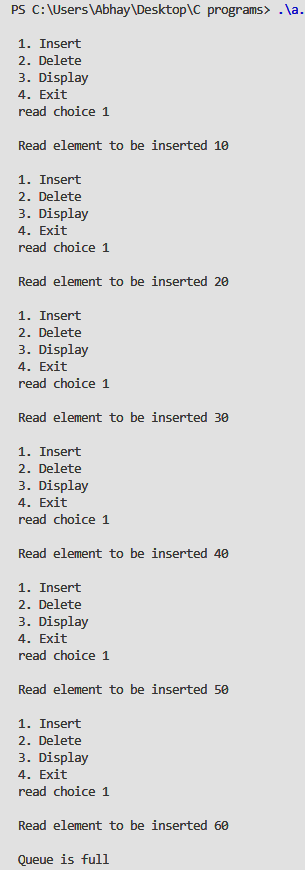
}

**Output:**

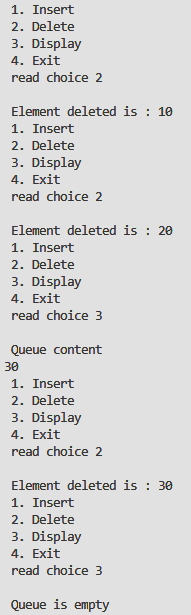
**Enqueue and Display**



**Queue is full**



**Deletion and printing the queue is empty**



**Exit Command**

