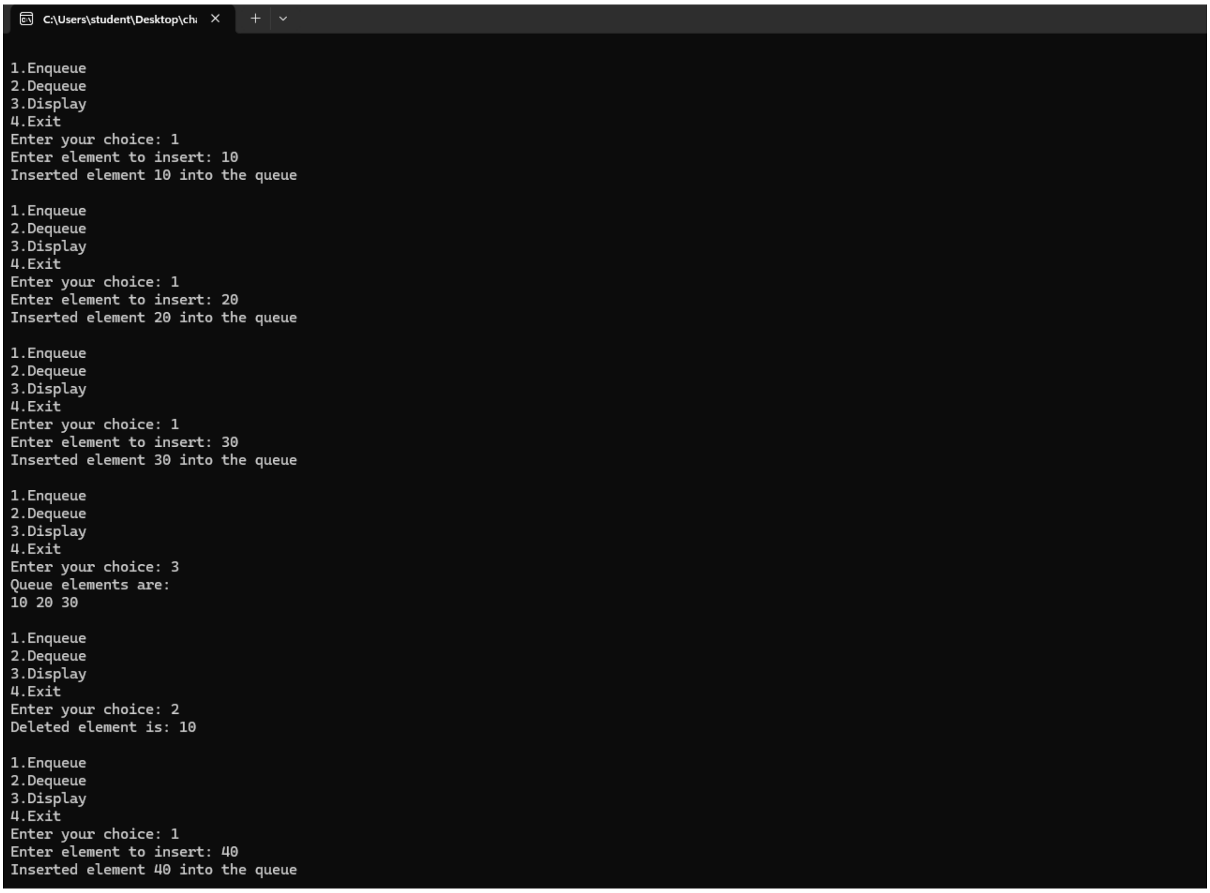
CIRCULAR QUEUE:

```
#include <stdio.h>
#define N 5

int queue[N];
int front=-1,rear=-1;

void enqueue(int x) {
if(front==-1 && rear==-1) {
front=rear=0;
queue[rear]=x;
printf("Inserted element %d into the queue\n",queue[rear]);
}
else if((rear+1)%N==front){
printf("Queue is full\n");
}
else {
rear=(rear+1)%N;
printf("Inserted element %d into the queue\n",x);
queue[rear]=x;
}
}

void dequeue() {
if (front==-1 && rear==-1){
```

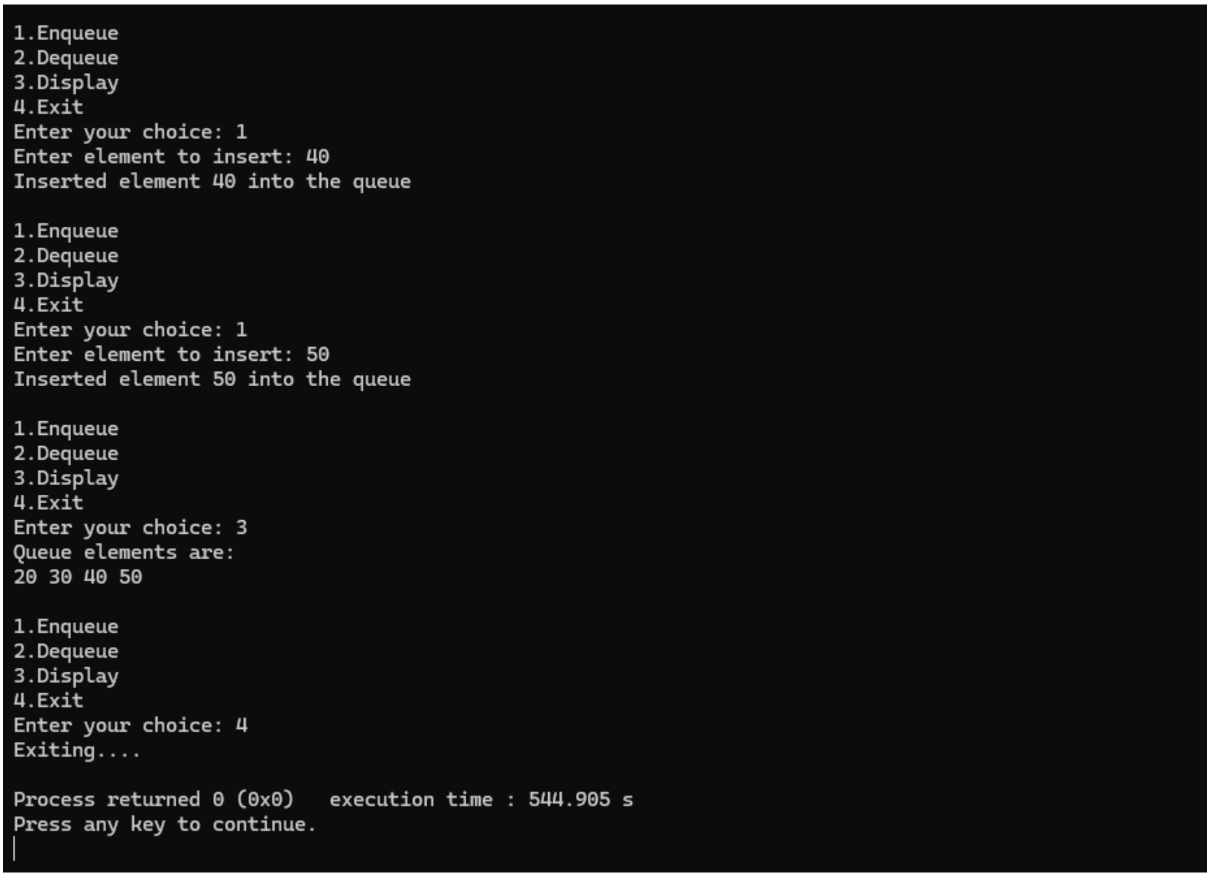
```
printf("Queue is empty\n");
}
else if(front==rear){
printf("Deleted element is:  %d\n",queue[front]);
front=rear=-1;
}
else{
printf("Deleted element is: %d\n",queue[front]);
front=(front+1)%N;
}
}

void display() {
if (front==-1 && rear==-1){
printf("Queue is empty\n");
}
else {
        printf("Queue elements are:\n");
        int i;
        for (i = front; i !=rear; i=(i + 1) % N)
            {
            printf("%d ", queue[i]);
        }
         printf("%d ", queue[i]);
        }
printf("\n");
```

```
}




int main() {
int choice,x;

do{
printf("\n1.Enqueue\n");
printf("2.Dequeue\n");
printf("3.Display\n");
printf("4.Exit\n");
printf("Enter your choice: ");
scanf("%d",&choice);

switch(choice) {
case 1:
   printf("Enter element to insert: ");
   scanf("%d",&x);
   enqueue(x);
   break;

case 2:
   dequeue();
   break;
case 3:
   display();
   break;

case 4:
   printf("Exiting....\n");
   break;

default:
   printf("Invalid Choice\n");
}
}
while (choice !=4);
return 0;
}
```

OUTPUT:



```
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter element to insert: 10
Inserted element 10 into the queue

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter element to insert: 20
Inserted element 20 into the queue

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter element to insert: 30
Inserted element 30 into the queue

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 3
Queue elements are:
10 20 30

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 2
Deleted element is: 10

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter element to insert: 40
Inserted element 40 into the queue
```



```
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter element to insert: 40
Inserted element 40 into the queue

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter element to insert: 50
Inserted element 50 into the queue

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 3
Queue elements are:
20 30 40 50

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 4
Exiting....

Process returned 0 (0x0)   execution time : 544.905 s
Press any key to continue.
```

OBSERVATION:

3/11/25

WAP to simulate the working of a circular queue of integers using array. Provide the following operations insert, delete ~~Algorithm:-~~ & display. The prg should print appropriate messages for queue empty

Algorithm:-

```
1. Start the program
2. Display an array of size N
3. Declare Queue [N]
4. Set front & rear = -1

5. To perform Enqueue /-Insert :-
-> If front == -1 && rear == -1,
      set front & rear = 0, then insert
   the element at Queue [rear].
-> Else if, (rear+1) % N = front, then
      print Queue is full
-> Else increment rear [(rear+1) % N] then
      insert x at Queue [rear].

6. To perform Dequeue/Delete :-
-> If front & rear = -1, then print
      Queue is Empty.
-> Else if, front = rear then print
      Deleted element from Queue, Queue[front]
   & set front & rear = -1
-> Else, print the deleted element &
   increment front [front+1] % N.

7. To Display :-
-> If front & rear = -1, then print that
```

Queue is Empty.

=> Else, print the elements in Queue
for i = front to rear, print Queue[i].

8. 3/11/25

```
for (i = front; i<=rear; i++)
{
    printf ("%d", queue[i]);
}
```

```
# include<stdio.h>
=> # define N5

int queue[N];
int front ==-1, rear==-1;

void enqueue(int x)
{
   if (front==-1 && rear==-1)
   {
       front = rear =0;
       queue[rear] = x;
       printf ("Inserted element %d into the
           queue\n", queue[rear]);
   }
   else if ((rear+1)%N == front)
   {
       printf("Queue is full\n");
   }
   else {
     rear = (rear+1) % N,
     printf ("Inserted element %d into the
       queue\n", x);
     queue[rear] = x;
   }
}
```

```
void dequeue()
 {
    if (front ==-1 && rear==-1)
     {
        printf("Queue is empty\n");
     }
 else if (front ==rear)
    {
       printf("Deleted element is: %d\n",
                  queue[front]);
       front = rear = -1;
    }
else {
    printf("Deleted element is: %d\n",
          queue[front]);
    front = (front+1)%N;
  }
}

void display()
 {
  if (front ==-1 && rear ==-1)
   {
      printf("Queue is empty\n");
   }
 else {
    printf("Queue elements are: \n");
    int i;
    for (i= front; i!= rear; i=(i+1)%N)
     {
        printf("%d ", queue[i]);
     }
    printf("%d ", queue[i]);
  }
```

```
printf("\n");
}

int main()
{
  int choice, x;

do {
   printf("\n1. Enqueue\n");
   printf(" 2. Dequeue \n");
   printf(" 3. Display \n");
    printf(" 4. Exit  \n");
   printf("Enter your choice : ");
    scanf("%d",&choice);

switch (choice)
 {
   case 1:
      printf("Enter element to insert:");
       scanf("%d",&x);
       enqueue(x);
        break;

   case 2:
      deque();
       break;

    case 3:
       display();
         break;

    case 4:
     printf("\nExiting.....\n");
     break;
default :
    printf ("Invalid choice \n");
  }
}

while (choice != 4);
 return 0;
}
```

Output :-

```
1. Enqueue
2. Deque
3. Display
4. Exit
Enter your choice : 1
Enter element to insert : 10
Inserted element 10 into the queue

1. Enqueue
2. Deque
3. Display
4. Exit
Enter your choice : ~~2~~ 1
Enter element to insert : 20
Inserted element 20 into the queue

1. Enqueue
2. Deque
3. Display
4. Exit
Enter your choice : 1
Enter element to insert : 30
Inserted element 30 into the equeue.

1. Enqueue
2. Deque
3. Display
4. Exit
   Enter your choice : 3
   Queue elements are :
     10 20 30

1. Enqueue
2. Dequeo
3. Display
4. Exit
   Enter your choice : 2
   Delement element is 10.

1. Enqueue
2. Deque
3. Display
4. Exit
   Enter your choice : 1
   Enter element to insert : 40
   Inserted element 40 into the queue.

1. Enqueue
2. Deque
3. Display
4. Exit
   Enter your choice : 1
   Enter element to insert : 50
   Inserted element 50 into the queue.
```

```
1  Enqueue
2  Deque
3. Display
4  Exit.
   Enter your choice : 3
   Queue elements are :
     20  30  40  50.
```

3/11/22