Date:

LAB 4

Page No.

```
#include <stdio.h>
#include <stdlib.h>
int front = -1;
int rear = -1;
const int MAX;
void Enqueue(int, int queue[]);
int Dequeue (int queue[]);
void display (int queue[]);
int main (int argc, char **argv)
{
 int size;
printf ("Enter the size of circular queue");
Scanf("%d", &MAX);
int queue[MAX];
int option;
int item;
do {
printf ("Circular Queue\n");
printf ("1. Enqueue\n");
printf ("2. Dequeue\n");
printf ("3. Display the Content\n");
printf ("4. Exit\n");
printf ("Enter the option :");
Scanf ("%d", &option);
switch (option)
```

Date:

Page No.

```
{
case1: printf ("Enter the element \n");
  scanf ("%d", &item);
  Enque( item, queue);
 break;
case 2: item = Dequeue (queue);
 if (item==-1)
printf ("Queue is empty\n");
 else
printf ("Removed element from the queue %d\n",
 item);
break;
case 3: display (queue);
 break;
Case 4: exit(0);
}
} while (option != 4);
return 0; }
void Enqueue (int ele, int queue[])
{
if (((front==0 && rear==MAX-1)) || (front==rear+1))
{
printf ("Queue is full \n");
return;
 }
else
```

Date:

Page No.

```
{  rear = (rear+1) % MAX;
queue[rear] = ele;
if(front == -1)
front = 0;
} }
int Dequeue (int queue[])
{
int item;
if ((front == -1) && (rear == -1))
{
return(-1);
}
else
{ item = queue[front];
if(front == rear)
{
front = -1;
 rear = -1;
} else
{
front = (front+1) % MAX;
} return item;
}
}
```

Date:

Page No.

```
void display(int queue[])
{
int i;
if(((front == -1) && (rear == -1)))
{
printf("Queue is empty\n"); return;
}
else if(front == rear)
{
printf("\n Queue contents :\n");
printf("%d", queue[front]);
printf("\n");
}
else if (rear > front)
{
printf("\n Queue contents : \n");
for(i=front; i<=rear; i++)
printf("%d", queue[i]);
printf("\n"); }
else if (front > rear)
{ printf("Queue contents: \n");
for(i=front; i<=MAX-1; i++)
printf("%d", queue[i]);
for(i=0; i<=rear; i++)
printf("%d", queue[i]);
printf("\n"); } }
```