WEEK 5

Lab Exercises:

1: Implement a circular queue of Strings using structures. Include functions insertcq, deletecq and displaycq.

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAXIMUM 5
#define STRING 5
typedef struct
{
char cqueue[MAXIMUM][STRING];
int front;
int rear;
} CQUEUE;
void insert(CQUEUE *queue,char a[])
{
if((queue->rear+1)%(MAXIMUM+1)==queue->front)
{
printf("\nQueue is full!!");
return;
}
queue->rear=(queue->rear+1)%(MAXIMUM+1);
strcpy(queue->cqueue[queue->rear],a);
}
void delete(CQUEUE *queue)
{
if(queue->rear==queue->front)
{
```

```
printf("\nQueue is empty!!");
return;
}
char ch[STRING];
queue->front=(queue->front+1)%(MAXIMUM+1);
strcpy(ch,queue->cqueue[queue->front]);
printf("%s ",ch);
}
void display(CQUEUE *queue)
{
if(queue->rear==queue->front)
{
printf(" \nQueue is empty!!");
return;
}
int myName =(queue->front+1)%(MAXIMUM+1);for(; myName!=queue->rear; myName =(myName+1)%(MAXIMUM+1))
{
printf("%s ",queue->cqueue[myName]);
}
printf("%s ",queue->cqueue[myName]);
}
int main()
{
char myName[MAXIMUM];
CQUEUE *queue,queue1;
queue=&queue1;
queue->rear=queue->front=-1;
while(1)
{
int choice;
printf("\nIMPLEMENTATION OF CIRCULAR QUEUE");
printf("\n1: INSERT");
printf("\n2: DELETE");
printf("\n3: DISPLAY");
```

```
printf("\n4: EXIT");
printf("\nEnter your choice: ");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("\nEnter the string: ");
scanf("%s",myName);
insert(&queue1,myName);
break;
case 2:
printf("\nQueue after deletion: ");
delete(&queue1);
break;
case 3:
printf("\nCurrent queue is:");
display(&queue1);
break;
case 4:
exit(0);
break;
default :
printf("\nInvalid Choice!!");
}
}
return 0;
}
```

Test Case:

```
IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter your choice: 1

Enter the string: abc

IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter your choice: 1

Enter the string: cde

IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter your choice: 3

Current queue is:abc cde
IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter your choice: 2

Queue after deletion: abc
IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter your choice: 2

Queue after deletion: cde
IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter your choice: 2

Queue after deletion:
Queue is empty!!
IMPLEMENTATION OF CIRCULAR QUEUE
1: INSERT
2: DELETE
3: DISPLAY
```

2: Implement two circular queues of integers in a single array where first queue will run from 0 to N/2 and second queue will run from N/2+1 to N-1 where N is the size of the array.

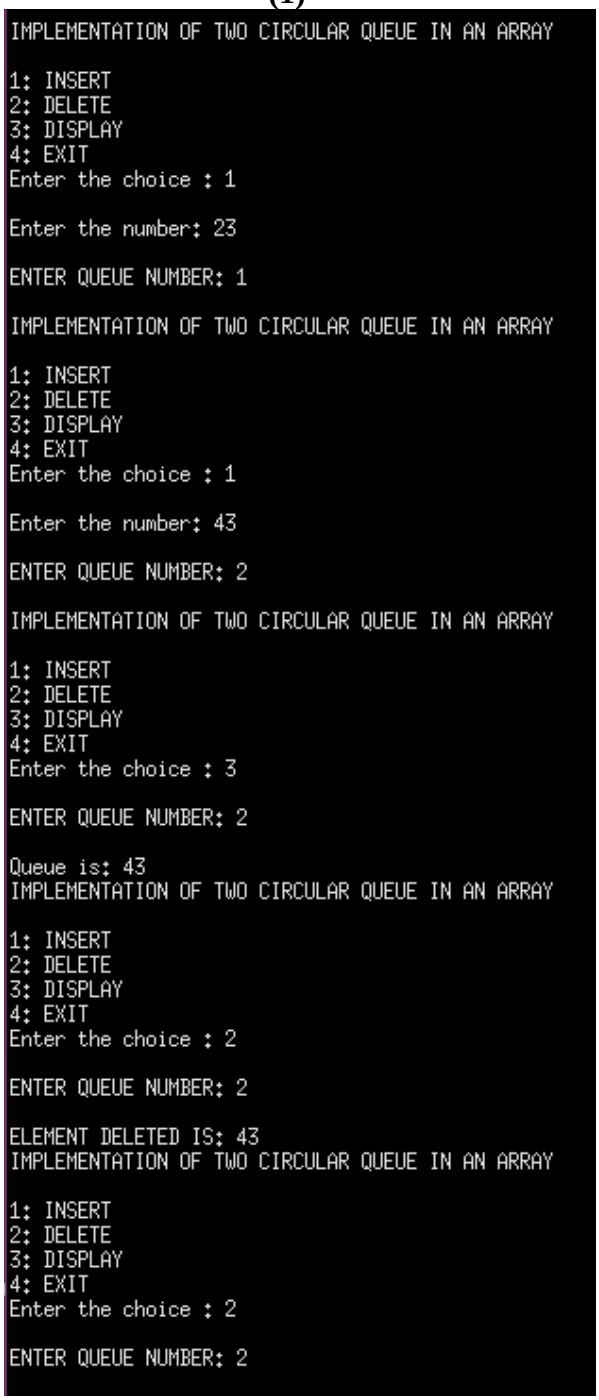
Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define SIZE 20
#define n 2
int MAXIMUM = SIZE/n;
int queue[SIZE];
int front[n];
int rear[n];
void construct(){
rear[0]=front[0]=0;
rear[1]=front[1]=SIZE/n;
}
void insert(int i,int ch){
if((rear[i]+1)%(MAXIMUM)+ (i*MAXIMUM)==front[i]){
printf("\ntQueue is empty!!");
return;
}
rear[i]=(rear[i]+1)%(MAXIMUM)+ (i*MAXIMUM);
queue[rear[i]]=ch;
}
int delete(int i){
if(rear[i]==front[i]){
printf("\nQueue is empty!");
return -1;
}
front[i]=(front[i]+1)%(MAXIMUM)+ (i*MAXIMUM);
return queue[front[i]];
}
void display(int i){
if(rear[i]==front[i]){
printf("\nQueue is empty!!");
return ;
}
int myName;
printf("\nQueue is: ");
for(myName=(front[i]+1)%MAXIMUM+(i*MAXIMUM);
myName!=rear[i];
myName=((myName+1)%MAXIMUM+(i*MAXIMUM)))printf("%d ",queue[myName]);
printf("%d ",queue[myName]);
}
int main(){
construct();
int myName,i,j;
while(1)
```

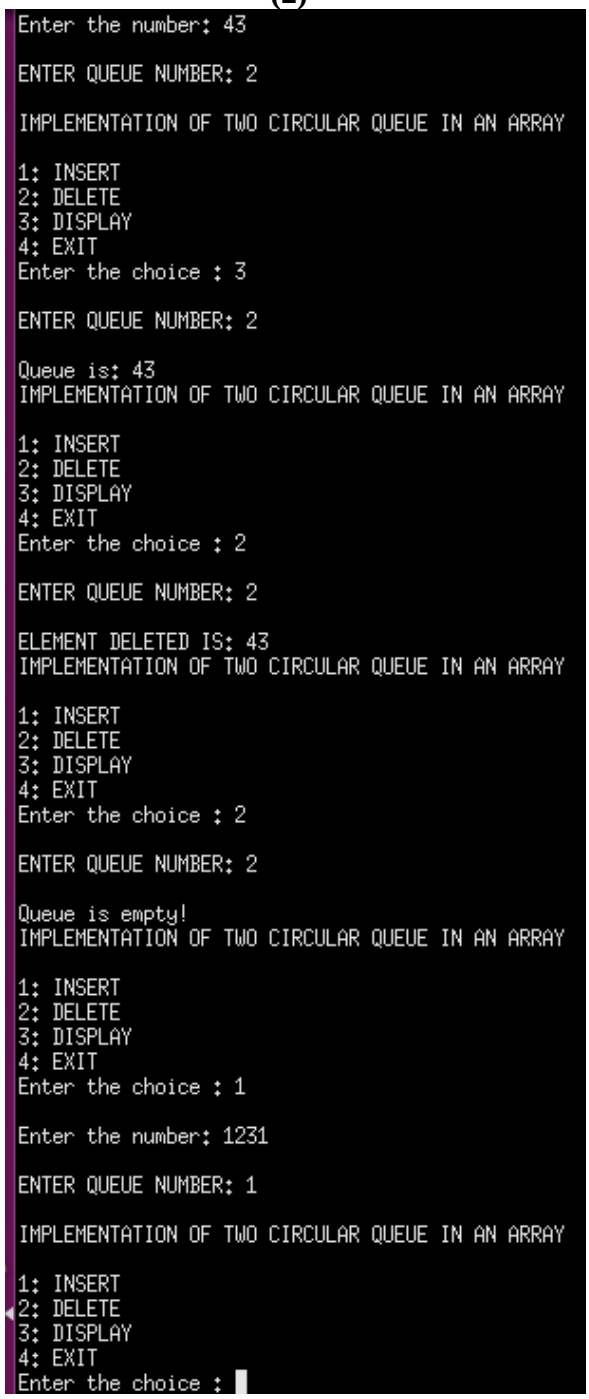
```
{
int choice;
printf("\nIMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY\n");
printf("\n1: INSERT");
printf("\n2: DELETE");
printf("\n3: DISPLAY");
printf("\n4: EXIT");
printf("\nEnter the choice : ");
scanf("%d",&choice);
switch(choice){
case 1:
printf("\nEnter the number: ");
scanf("%d",&myName);
printf("\nENTER QUEUE NUMBER: ");
scanf("%d",&i);
insert(i-1,myName);
break;
case 2:
printf("\nENTER QUEUE NUMBER: ");
scanf("%d",&i);
j = delete(i-1);
if(j!=-1)printf("\nELEMENT DELETED IS: %d",j);
break;
case 3:
printf("\nENTER QUEUE NUMBER: ");
scanf("%d",&i);
display(i-1);
break;
case 4 :
exit(0);
default:
printf("\nInvalid Choice!!");
break;
}}
return 0;
}
```

Test Case:

(1)


IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 1
Enter the number: 23
ENTER QUEUE NUMBER: 1
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 1
Enter the number: 43
ENTER QUEUE NUMBER: 2
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 3
ENTER QUEUE NUMBER: 2
Queue is: 43
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 2
ENTER QUEUE NUMBER: 2
ELEMENT DELETED IS: 43
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 2
ENTER QUEUE NUMBER: 2

(2)


Enter the number: 43
ENTER QUEUE NUMBER: 2
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 3
ENTER QUEUE NUMBER: 2
Queue is: 43
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 2
ENTER QUEUE NUMBER: 2
ELEMENT DELETED IS: 43
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 2
ENTER QUEUE NUMBER: 2
Queue is empty!
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice : 1
Enter the number: 1231
ENTER QUEUE NUMBER: 1
IMPLEMENTATION OF TWO CIRCULAR QUEUE IN AN ARRAY
1: INSERT
2: DELETE
3: DISPLAY
4: EXIT
Enter the choice :

3: Implement a queue with two stacks without transferring the elements of the second stack back to stack one. (use stack1 as an input stack and stack2 as an output stack).

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAXSIZE 10
int input[MAXSIZE];
int output[MAXSIZE];
int top1=-1,top2=-1;
void insert(int i){
if(top1==MAXSIZE-1){
printf("\nQUEUE IS FULL!!");
return;
}
input[++top1]=i;
}
int delete(){
if(top1==-1 && top2==-1){
printf("\nQUEUE IS EMPTY!!");
return -1;
}
if(top2==-1){
while(top1!=-1){
output[++top2]=input[top1--];
}
}
return output[top2--];
}
void display(){
if(top1==-1 && top2==-1){
printf("\nQUEUE IS EMPTY!!");
return ;
}
```

```
int myName = top2;
printf("\nQUEUE IS: ");
for(;myName>-1;myName--)
printf("\n%d ",output[myName]);
for(myName=0;myName<=top1;myName++)
printf("\n%d ",input[myName]);
}
int main(){
int x,i,j;
while(1)
{
int choice;
printf("\nIMPLEMENTATION OF QUEUE WITH TWO STACK");
printf("\n1: INSERT");
printf("\n2: DELETE");
printf("\n3: DISPLAY");
printf("\n4: EXIT");
printf("\nEnter your choice : ");
scanf("%d",&choice);
switch(choice){
case 1:
printf("\nENTER THE ELEMENT: ");
scanf("%d",&x);
insert(x);
break;
case 2:
j = delete();
if(j!=-1) printf("\nELEMENT DELETED IS: %d",j);
break;
case 3:
display();
break;
case 4 :
exit(0);
default:
```

```
printf("\nInvalid choice!!");
break;
}
}
return 0;
}
```

Test Case: