

LAB PROGRAM 3B

```
#include<stdio.h>

#define MAX 5

int queue[MAX];

int front=-1, rear=-1;

void insert(int value)

{

if((front==0&&rear==MAX-1)||((front==(rear+1)%MAX))

{

    printf("Queue Overflow!\n", value);

}

else

{

    if(front==-1)

    {

        front=0;

        rear=0;

    }

    else

    {

        rear=(rear+1)%MAX;

    }

    queue[rear]=value;

    printf("%d inserted into the queue\n", value);
```

```
}  
  
}  
  
void delete()  
{  
  
    if(front==-1)  
    {  
        printf("Queue Underflow");  
    }  
    else  
    {  
        printf("Deleted element: %d\n", queue[front]);  
        if(front==rear)  
        {  
            front== -1;  
            rear== -1;  
        }  
        else  
        {  
            front=(front+1)%MAX;  
        }  
    }  
}  
  
void display()
```

```

{
    if(front==-1)
    {
        printf("Queue is empty\n");
    }
    else
    {
        printf("Queue elements: ");
        int i=front;
        while(1)
        {
            printf("%d", queue[i]);

            if(i==rear)
                break;

            i=(i+1)%MAX;
        }
        printf("\n");
    }
}

int main()
{
    int choice, value;

    while(1)
    {

```

```
printf("\nCircular Queue Operations\n");
```

```
printf("1. Insert\n");
```

```
printf("2. Delete\n");
```

```
printf("3. Display\n");
```

```
printf("4. Exit\n");
```

```
printf("Enter your choice: ");
```

```
scanf("%d", &choice);
```

```
switch(choice)
```

```
{
```

```
case 1:
```

```
    printf("Enter value to insert: ");
```

```
    scanf("%d", &value);
```

```
    insert(value);
```

```
    break;
```

```
case 2:
```

```
    delete();
```

```
    break;
```

```
case 3:
```

```
    display();
```

```
    break;
```

```
case 4:
```

```
    printf("Exiting program\n");
```

```
    return 0;
```

```

default:

    printf("Invalid choice\n");

}

}

return 0;

}

```

OUTPUT:

```

Circular Queue Operations
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 34
34 inserted into the queue

Circular Queue Operations
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 22
22 inserted into the queue

Circular Queue Operations
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 34

Circular Queue Operations
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 22

Circular Queue Operations
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting program

Process returned 0 (0x0)   execution time : 12.461 s
Press any key to continue.

```