main.c

```c
#include <stdio.h>
#include <stdlib.h>

#define MAX 3

void insert();
void delete();
void display();
int queue_array[MAX];11
int rear = - 1;
int front = - 1;
int main()
{
    int choice;
    while (1)
    {
        printf("1.Insert element to queue \n");
        printf("2.Delete element from queue \n");
        printf("3.Display all elements of queue \n");
        printf("4.Quit \n");
        printf("Enter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
            insert();
```

SVA_DOA.txt

nnyEarlySphere

Run

main.c

```
25                  case 1:
26                  insert();
27                  break;
28                  case 2:
29                  delete();
30                  break;
31                  case 3:
32                  display();
33                  break;
34                  case 4:
35                  exit(1);
36                  default:
37                  printf("Wrong choice \n");
38              }
39          }
40      }
41
42      void insert()
43      {
44          int add_item;
45          if (rear == MAX - 1)
46          printf("Queue Overflow \n");
47          else
48          {
49              if (front == - 1)
50
```

main.c

```
45      if (rear == MAX - 1)
46      printf("Queue Overflow \n");
47      else
48      {
49          if (front == - 1)
50
51          front = 0;
52          printf("Inset the element in queue : ");
53          scanf("%d", &add_item);
54          rear = rear + 1;
55          queue_array[rear] = add_item;
56      }
57  }
58
59  void delete()
60  {
61      if (front == - 1 || front > rear)
62      {
63          printf("Queue Underflow \n");
64          return ;
65      }
66      else
67      {
68          printf("Element deleted from queue is : %d \n",
            queue_array[front]);
69          front = front + 1;
```

```
66          else
67          {
68              printf("Element deleted from queue is : %d\n",
                queue_array[front]);
69              front = front + 1;
70          }
71      }
72
73      void display()
74      {
75          int i;
76          if (front == - 1)
77              printf("Queue is empty \n");
78          else
79          {
80              printf("Queue is : \n");
81              for (i = front; i <= rear; i++)
82                  printf("%d ", queue_array[i]);
83              printf("\n");
84          }
85      }
86
87
```

```
$ clang-7 -pthread -lm -o main main.c
$ ./main
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 23
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 22
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 20
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 45
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
```

```
4.Quit
Enter your choice : 1
Inset the element in queue : 45
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
23 22 20 45
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 23
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 22
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 20
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
```

```c
#include <stdio.h>
#include <stdlib.h>
#define SIZE 3
int a[SIZE],t;
int front=-1;
int rear=-1;

int IsEmpty()
{
 if(rear==-1 && front==-1)
   return 1;
  else
   return 0;
}


int IsFull()
{
 if(front==(rear+1)%SIZE)
 return 1;
 else
return 0;
}


void Enqueue(int x)
```

main.c

```
27  {
28      if(IsFull())
29          printf("The queue is full\n");
30      else if(IsEmpty())
31      {
32          front=0;
33          rear=0;
34          a[rear]=x;
35      }
36      else
37      {
38              rear=(rear+1)%SIZE;
39              a[rear]=x;
40      }
41  }
42
43  int Dequeue()
44  {     int x;
45      if(IsEmpty())
46      printf("The queue is empty.\n");
47      else if(front==rear)
48      {
49          x=a[front];
50          front=-1;
51          rear=-1;
52           printf("The element was removed\n");
```

main.c

```
53      }
54      else
55      {
56          x=a[front];
57          front=(front+1)%SIZE;
58          printf("The element was removed\n");
59      }
60      return x;
61  }
62
63
64  void display()
65
66  {
67      if (front == -1)
68      {
69          printf("\nQueue is Empty");
70          return;
71      }
72      printf("\nElements in Circular Queue are:\n");
73      if (rear >= front)
74      {
75          for (int i = front; i <= rear; i++)
76              printf("%d\n",a[i]);
77      }
78      else
79      {
```

main.c

```
78      else
79      {
80          for (int i = front; i < SIZE; i++)
81              printf("%d\n", a[i]);
82
83          for (int i = 0; i <= rear; i++)
84              printf("%d\n", a[i]);
85      }
86  }
87
88
89
90  int main()
91  {
92   int n,a;
93   while(1)
94    {
95   printf("Enter the
     operation.\n1-Insert\n2-Delete\n3-Display\n4-Exit\n");
96   scanf("%d",&n);
97   switch(n)
98   {
99     case 1: printf("Enter the element\n");
100              scanf("%d",&a);
101              Enqueue(a);
102               break;
```

RequiredIncredibleSy...

Stop ■

main.c

```
while(1)
 {
printf("Enter the
operation.\n1-Insert\n2-Delete\n3-Display\n4-Exit\n");
scanf("%d",&n);
switch(n)
{
  case 1: printf("Enter the element\n");
                scanf("%d",&a);
                Enqueue(a);
                 break;

  case 2 :    Dequeue();

                break;
  case 3:  display();
              break;
   case 4: exit(0);
    default :    printf("There is no such operation\n");
}
}
return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
1
Enter the element
15
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
2
The element was removed
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
1
Enter the element
23
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
3
```

```
Enter the element
15
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
2
The element was removed
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
1
Enter the element
23
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
3

Elements in Circular Queue are:
23
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
```