Circular Queue

Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 4

void Insert ();
int Delete ();
void Display ();

int cq[20]; int front=-1, rear=-1, item, ch, i;

void main ()
{
        while (1)
        {
                printf("\n 1. Insert \n 2. Delete \n 3. Display ~~\n 4. Exit~~
                        \n 4. Exit ");
                printf("\n Enter Your Choice: ");
                scanf("%d", &ch);
                switch (ch)
                {
                        case 1: Insert ();
                                break;
                        case 2: ~~Delete~~ item= Delete();
                                if (item != -1) {
                                        printf ("The Dequed Element is : %d,
                                                item);
                                }
                break;
        Case 3: display ();
                break;
        case 4: exit(0);
      }
    }
}

void Insert ()
{
    if (front == (rear +1) % MAX)
    {
        printf("Circular Queue is Full. \n");
        return;
    }
    if (rear ==-1 && front ==-1)
    {
        rear=0;
        front=0;
    }
    else
    {
        rear= (rear+1) % MAX;
    }
    printf("Enter Element to be Inserted : ");
    scanf(" %d", &item);
    cq[rear]= item;
    return;
}
```

PAGE NO :
DATE :

```
int Delete()
{
        if (front==-1 && rear==-1)
        {
                printf("Circular Queue is Empty \n");
                return(-1);
        }
        item= cq[front];
        if (front==rear)
        {
                front=-1;
                rear=-1;
        }
        else
        {
                front=(front+1) % MAX;
        }
        return item;

void Display()
{
        if (front==-1 && rear==-1)
        {
                printf("Circular Queue is Empty. \n");
                return;
        }
        printf("Circular Queue Contents: \n");
        if (front <= rear)
        {
                for (int i=0; i<=rear; i++){
                        printf("%d\n", cq[i]);
                }
        }
```

```
    else
    {
        for(int i= front; i<= MAX-1 ; i++)
        {
            printf("%d \n", cq[i]);
        }
        for(int i=0; i<=rear; i++)
        {
            printf("%d \n", cq[i]);
        }
    }
    return;
}
```

Output

```
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice : 1
Enter the Element to be Inserted: 10

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:1
Enter the Element to be Inserted : 20

1. Insert
2. Delete
3. Display
```

```
4. Exit
Enter Your Choice: 1
Enter the Element to be Inserted: 30

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Enter the Element to be Inserted : 40

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Circular Queue is Full

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 3
Circular Queue Contents:
10
20
30
40

1. Insert
2. Delete
3. Display
```

[illegible]

```
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 10

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequed Element is: 20

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 30

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued Element is. 40

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
Circular Queue is Empty.
```

PAGE NO :

DATE :

```
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 4
```

[illegible]