Data Structures Week 4:

Name: Aaryan Prakash

USN: 1BM23SC006

Class: 3A

Question: 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(" \n1. Insert \n2. Delete \n3. Display \n4. Exit");
      printf("\nEnter Your Choice: ");
      scanf("%d",&ch);
      switch(ch)
      {
         case 1: Insert();
```

```
            break;
        case 2: item=Delete();
            if (item!=-1)
            {
               printf("The Dequeued 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 the Element to be Inserted: ");
   scanf("%d",&item);
   cq[rear]=item;
```

```
    return;
}

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=front;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
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 10
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued 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.

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

Process returned 0 (0x0)   execution time : 43.123 s
Press any key to continue.
```