Data Structures

Experiment no. 3

Develop code to implement Linear & Circular Queue

Q. WAP in C to implement Linear Queue.

Code:

```
#include<stdio.h>
#include<conio.h>
#define Max 5

int queue[Max];
int f=-1;
int r=-1;

void Enqueue(int num)
{
        if(r==Max-1)
        {
                printf("\nQueue is Full\n");
        }
        else if(f==-1 && r==-1)
        {
                f++;
                r++;
                queue[r]=num;
        }
        else
        {
          r++;
          queue[r]=num;
        }
}

void Dequeue()
{
        if(f==-1 || f>r)
        {
                printf("\nQueue is Empty\n");
        }
        else
        {
```

```
                printf("\nThe no. is : %d\n",queue[f]);
                f=f+1;
        }
}

void display()
{
        if(f==-1 || f>r)
        {
                printf("\nQueue is Empty\n");
        }
        else
        {
                int i;
                printf("\n");
                for(i=f;i<=r;i++)
                {
                        printf("%d\t",queue[i]);
                }
        }
}

void main()
{
        int num,ch;
        clrscr();
        do
        {
                printf("\n1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n");
                printf("\tEnter your choice : ");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("\nEnter no to be put in queue : ");
                                scanf("%d",&num);
                                Enqueue(num);
                                break;
                        case 2:
                                Dequeue();
                                break;
                        case 3:
                                display();
                                break;
                        case 4:
                                exit(0);
                        default:
                                printf("Enter valid choice!!\n");
```

```
                }
        }while(ch!=4);
        getch();
}
```

Output:

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 12

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 10

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 8

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 6

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 4

1.Enqueue
2.Dequeue
3.Display
```

```
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 2

Queue is Full

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

12      10      8       6       4
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 2

The no. is : 12

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

10      8       6       4
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 12

Queue is Full

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

10      8       6       4
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 2
```

```
The no. is : 10

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

8       6       4
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 10

Queue is Full

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

8       6       4
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 4
```

Q. WAP in C to implement Circular Queue.

Code:

```
#include<stdio.h>
#include<conio.h>
#define Max 5

int queue[Max];
int f=-1;
int r=-1;

void Enqueue(int num)
{
	if(f==-1 && r==-1)
	{
		f=0;
		r=0;
		queue[r]=num;
	}
	else if((r+1)%Max==f)
	{
		printf("\nThe Queue is full\n");
	}
	else
	{
		r=(r+1)%Max;
		queue[r]=num;
	}
}

void Dequeue()
{
	if(f==-1 && r==-1)
	{
		printf("\nQueue is Empty\n");
	}
	else if(f==r)
	{
		f=-1;
		r=-1;
	}
	else
	{
		printf("\nDeleted Element of the Queue : %d",queue[f]);
		f=(f+1)%Max;
	}
}
```

```
void display()
{
	int i;
	printf("\n");
	if (f>r)
	{
		for (i = f; i < Max; i++)
		{
			printf("%d ", queue[i]);
		}
		for (i = 0; i <= r; i++)
		{
			printf("%d ", queue[i]);
		}
	}
	else
	{
		for (i = f; i <= r; i++)
		{
			printf("%d ", queue[i]);
		}
	}
}

void main()
{
	int num,ch;
	clrscr();
	do
	{
		printf("\n1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n");
		printf("\tEnter your choice : ");
		scanf("%d",&ch);
		switch(ch)
		{
			case 1:
				printf("\nEnter no to be put in queue : ");
				scanf("%d",&num);
				Enqueue(num);
				break;
			case 2:
				Dequeue();
				break;
			case 3:
				display();
				break;
			case 4:
```

```
                exit(0);
            default:
                printf("Enter valid choice!!\n");
        }
    }while(ch!=4);
    getch();
}
```

Output:

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 100

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 90

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 80

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 70

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 60
```

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 50

The Queue is full

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

100 90 80 70 60
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 2

Deleted Element of the Queue : 100
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

90 80 70 60
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 100

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

90 80 70 60 100
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 2
```

```
Deleted Element of the Queue : 90
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 1

Enter no to be put in queue : 90

1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 3

80 70 60 100 90
1.Enqueue
2.Dequeue
3.Display
4.Exit
        Enter your choice : 4
```