

## WEEK 4: circular queue

```
#include<stdio.h>

#define MAX 10

int cir_queue[MAX];
int front = -1;
int rear = -1;
void insert(int item)
{
    if((front == 0 && rear == MAX-1) || (front == rear+1))
    {
        printf("Queue Overflow \n");
        return;
    }
    if (front == -1)
    {
        front = 0;
        rear = 0;
    }
    else
    {
        if(rear == MAX-1)
            rear = 0;
        else
            rear = rear+1;
    }
    cir_queue[rear] = item ;
}

void delete()
```

```

{
    if (front == -1)
    {
        printf("Queue Underflow\n");
        return ;
    }
    printf("Element deleted from queue is : %d\n",cir_queue[front]);
    if(front == rear)
    {
        front = -1;
        rear=-1;
    }
    else
    {
        if(front == MAX-1)
            front = 0;
        else
            front = front+1;
    }
}

void display()
{
    int fpos = front,rpos = rear;
    if(front == -1)
    {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue elements :\n");
    if( fpos <= rpos )
        while(fpos <= rpos)

```

```

        {
            printf("%d ",cir_queue[fpos]);
            fpos++;
        }
else
{
    while(fpos <= MAX-1)
    {
        printf("%d ",cir_queue[fpos]);
        fpos++;
    }
    fpos = 0;
    while(fpos <= rpos)
    {
        printf("%d ",cir_queue[fpos]);
        fpos++;
    }
}
printf("\n");
}

int main()
{
    int select,item;
    do
    {
        printf("1.Insert\n");
        printf("2.Delete\n");
        printf("3.Display\n");
        printf("4.Terminate\n");

        printf("Enter selection: ");

```

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

switch(select)
{
    case 1 :
        printf("Insert element : ");
        scanf("%d", &item);

        insert(item);
        break;
    case 2 :
        delete();
        break;
    case 3:
        display();
        break;
    case 4:
        break;
    default:
        printf("Wrong choice\n");
}
}while(select!=4);

return 0;

}
```

```
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 1
Insert element : 2
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 1
Insert element : 3
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 1
Insert element : 4
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 3
Queue elements :
2 3 4
```

```
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 2
Element deleted from queue is : 2
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 3
Queue elements :
3 4
1.Insert
2.Delete
3.Display
4.Terminate
Enter selection: 4
```