

```
//Queue Liste Implementasyonu
```

```
#include<iostream>
using namespace std;
```

```
struct node {
    int data;
    node *next;
};
```

```
struct queue {
    int cnt;
    node *front;
    node *rear;
};
```

```
node *enqueue(queue *qu, int x) {
```

```
    if (qu->cnt == 0)
    {
        qu->rear = new node;
        qu->rear->data = x;
        qu->rear->next = NULL;
        qu->cnt++;
        qu->front = qu->rear;
    }
```

```
    else
    {
        qu->rear->next = new node;
        qu->rear = qu->rear->next;
        qu->rear->data = x;
        qu->rear->next = NULL;
        qu->cnt++;
    }
```

```
    return qu->front;
}
```

```
node *dequeue(queue *qu) {
```

```
    if (qu->cnt == 0)
        cout << "kuyruk bos";
```

```
    if (qu->cnt == 1)
    {
        node *yrd;
        yrd = qu->front;
        qu->front = NULL;
        qu->rear = NULL;
        qu->cnt--;
        //yrd->data kullanılacak;
        delete yrd;
    }
```

```
    else
    {
        node *yrd;
        yrd = qu->front;
        qu->front = yrd->next;
        yrd->next = NULL;
        //yrd->data kullanılacak;
        delete yrd;
        qu->cnt--;
    }
```

```
    return qu->front;
}
```

```
void yazdir(queue *qu) {
```

```
    if (qu->cnt == 0)
        cout << "kuyruk bos";
    else
    {
        node *yrd;
        yrd = qu->front;
        while (yrd != NULL)
        {
            cout << yrd->data << " ";
            yrd = yrd->next;
        }
        cout << endl;
    }
    system("pause");
}
```

```
void main() {
```

```
    queue *q;
    q = new queue;
    q->cnt = 0;
    q->front = NULL;
    q->rear = NULL;
```

```
    enqueue(q, 10);
    enqueue(q, 20);
    enqueue(q, 30);
    enqueue(q, 440);
    enqueue(q, 550);
    yazdir(q);
    dequeue(q);
    dequeue(q);
    dequeue(q);
    yazdir(q);
    enqueue(q, 630);
    enqueue(q, 740);
    enqueue(q, 850);
    yazdir(q);
}
```