

a) Aşağıdaki diziden elde edilen alist adlı bir list nesnesi bildiriniz.

```
int arr[]={5,2,3,8,9,25,35};  
int arrSize = sizeof(arr)/sizeof(int);
```

YANIT:

```
list<int> alist(arr, arr+arrSize);
```

b) **iter** adlı bir list tekrarlayıcısı bildiriniz.

YANIT:

```
list<int>::iterator iter
```

c) listenin ilk ögesini işaret eden **iter** adlı bir tekrarlayıcıyı bildiriniz.

YANIT:

```
iter = alist.begin();
```

d) c şıkkının devamı olarak

```
iter++;
```

deyimi sonunda ***iter** değeri nedir?

YANIT: 2

e) **iter = alist.end();**

```
iter--;
```

```
iter--;
```

deyimleri sonunda ***iter** değeri nedir?

YANIT: 25

ADI SOYADI :

NUMARASI :

NOT :

BAŞKENT ÜNİVERSİTESİ ELEKTRİK ELEKTRONİK
MÜHENDİSLİĞİ BÖLÜMÜ

EEM 227 KISA SINAV 3 – 12 KASIM 2007

Aşağıda verilen **find_last_of** yöntemi tek sayıda elemandan oluşmuş simetrik bir kuyruğu(Queue) ve kuyruk boyunu argüman olarak alır. Simetrinin bozulduğu ilk noktanın konumunu (orta noktayı) döndürür. Buna göre aşağıdaki kodlarda boş yerleri doldurunuz: Simetrik : {16, 16, 73, 16, 16} Simetrik değil : { 15, 15, 15, 73, 15, **19**, 15}

```
template <typename T>  
T find_last_of(queue<T>& alist, const T& length)  
{  
    T size=length;  
    do  
    {  
        alist.pop();  
        size--;  
    }  
    while( (alist.back()==alist.front()) );  
    return length-size;  
}
```

Aşağıda verilen **find_last_of** yöntemi bir listeyi ve bir değeri argüman olarak alır, ikinci argümanın liste içinde son tekrar ettiği yere (eğer varsa) bir tekrarlayıcı (iterator) döndürmektedir. Buna göre aşağıdaki kodlarda boş yerleri doldurunuz:

```
template <typename T>  
list<T>::iterator find_last_of(list<T>& alist, const T& value)  
{  
    list<T>::iterator iter = alist.end();  
    do  
        iter--;  
    while(iter != alist.begin() && !( *iter == value ));  
    return iter;  
}
```