



CEVAPLAR

```
void DoublyLinkedList::Bordo()
{
    header->next->next->prev = header;
    trailer->prev->next = header->next;
    header->next->prev = trailer->prev;
    trailer->prev = header->next;
    header->next = header->next->next;
    trailer->prev->next = trailer;
}

void main()
{
    DoublyLinkedList list;

    list.addBack("Omer");
    list.addBack("Oguzhan");
    list.addBack("Fatih");
    list.addBack("Ali Osman");

    list.Bordo();
    list.print();
}
```

Oguzhan
Fatih
Ali Osman
Omer

1. Yukarıdaki programın çıktısı nedir? (25 P)

```
void DoublyLinkedList::Mavi(DoublyNode* node)
{
    if (node->next == trailer)
    {
        cout << node->elem << endl;
    }
    else Mavi(node->next);
}

void main()
{
    DoublyLinkedList list;

    list.addBack("Omer");
    list.addBack("Oguzhan");
    list.addBack("Fatih");
    list.addBack("Ali Osman");

    list.Mavi(list.header);
}
```

Ali Osman

2. Yukarıdaki programın çıktısı nedir? (25 P)

```
int linearF(int a, int b, int c, int n)
{
    if (n <= 2) return c;
    else return linearF(b, c, a+b+c, n-1);
}

int tripleF(int k)
{
    if (k <= 2) return k;
    return tripleF(k-1) + tripleF(k-2) + tripleF(k-3);
}

void main()
{
    int linF = linearF(0, 1, 2, 7);
    cout << "linearF = " << linF << endl;

    int tripF = tripleF(7);
    cout << "tripleF = " << tripF << endl;
}
```

linearF = 37
tripleF = 37

3. tripleF() isimli fonksiyonun lineer recursive eşdeğeri olacak şekilde linearF()te ile gösterilen yerlere gerekli kodları yazınız. (25 P)

```
int binarySum(int A[], int i, int n)
{
    if (n == 1) return A[i];
    else
    {
        int Sum = binarySum(A, i, n/2 ) +
                binarySum(A, i + n/2, n/2 );
        cout << "Sum = " << Sum << endl;
        return Sum;
    }
}

void main()
{
    int A[8] = {1,2,3,4,5,6,7,8};

    binarySum(A, 0, 8);
}
```

Sum = 3
Sum = 7
Sum = 10
Sum = 11
Sum = 15
Sum = 26
Sum = 36

4. Yukarıdaki programın çıktısı nedir? (25 P)