



NUMARA :

AD SOYAD :

. ÖĞRETİM

DEĞERLENDİRME :

```
if(p->left == NULL && p->right != NULL)
{
    if(parent->left == p)
        parent->..... = p->.....;
    else
        parent->..... = p->.....;
    p->right = NULL;
    delete p;
}
if( p->left != NULL && p->right == NULL)
{
    if(parent->left == p)
        parent->..... = p->.....;
    else
        parent->..... = p->.....;
    p->left = NULL;
    delete p;
}
```

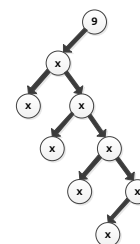
```
void addFront(const string& e)
{
    add(header->next, e);
}
void addBack(const string& e)
{
    add(trailer, e);
}
void add(DoublyNode* v, const string& e)
{
    DoublyNode* u = new DoublyNode;
    u->elem = e;
    u->next = v;
    u->prev = v->prev;
    v->prev->next = u;
    v->prev = u;
}
```

1. İkili ağaçtan eleman silen yukarıdaki kodda boş bırakılan yerleri doldurunuz. (20P)
2. Dairesel (circularly) bağlı liste, kuyruk (queue) olarak kullanılırken veri silme kodu neden aynıdır? (30P)

3. Çift yönlü bağlı listeye veri ekleyen yukarıda kodda addFront() fonksiyonu add()’i header->next ile çağırdığı halde addBack() neden trailer->prev değil de sadece trailer ile çağırmıştır? (20P)

8 7 6 5 4 3 2 1 9

4. Yukarıdaki verileri splay ağacına yerleştiriniz? (30P)
İpucu → Ağaç aşağıdaki gibi olmalıdır.



Bu soruyu kağıdın arka yüzüne çözünüz →