



<b>NUMARA :</b> .....	<b>AD SOYAD :</b> .....	<b>DEĞERLENDİRME</b>	
Sınavda Uyulması Gereken Kurallar	<b>İMZA :</b> .....	[.....]	.....
1. Cep telefonlarının saate bakmak için bile olsa herhangi bir amaçla kullanılması yasaktır. Telefon kapalı ve cepte olmalıdır. 2. Sınavın başında sorular kısaca açıklanacaktır. Öğrencilerin soruları cevaplandıktan sonra sınav boyunca soru sormak yasaktır. 3. Soru kağıdına numaranızı ve isminizi yazıp imzalamayı unutmayınız.			

```
void print1(DoublyNode* node)
{
    cout << node->elem << endl;
    if (node->next == trailer) return;
    else print1(node->next);
}

void print2(DoublyNode* node)
{
    if (node == trailer) return;
    else print2(node->next);
    cout << node->elem << endl;
}

void main()
{
    DoublyLinkedList list;
    list.insertOrdered("Paul", 720); // küçükten
    list.insertOrdered("Rose", 590); // büyüğe
    list.insertOrdered("Anna", 660); // sıralı ekle

    list.print1(list.header->next); // (1)
    list.print2(list.header->next); // (2)
    list.print3(list.trailer->prev); // (3)
    list.print4(list.trailer->prev); // (4)
}
```

```
void removeOrdered(const string& e, const int& i)
{
    DoublyNode* current = header->next;

    while (current != trailer)
    {
        if((current->elem == e) && (current->score == i))
        {
            current->prev->next = current->next;

            current->next->prev = current->prev;

            delete current;
            return;
        }
        current = current->next;
    }

    cout << e << " is not found" << endl;
}
```

1. a) print1() ve print2() fonksiyonları main() fonksiyonda header->next ile çağrıldıklarında çıktıları ne olur? (10P)

print1()	Print2()
Rose	Paul
Anna	Anna
Paul	Rose

b) main() fonksiyonda trailer->prev ile çağrıldıklarında print1() ile aynı çıktıyı verecek print3() fonksiyonunu; print2() ile aynı çıktıyı verecek print4() fonksiyonunu yazınız. İpucu → Cevaplar 3 'er satır.

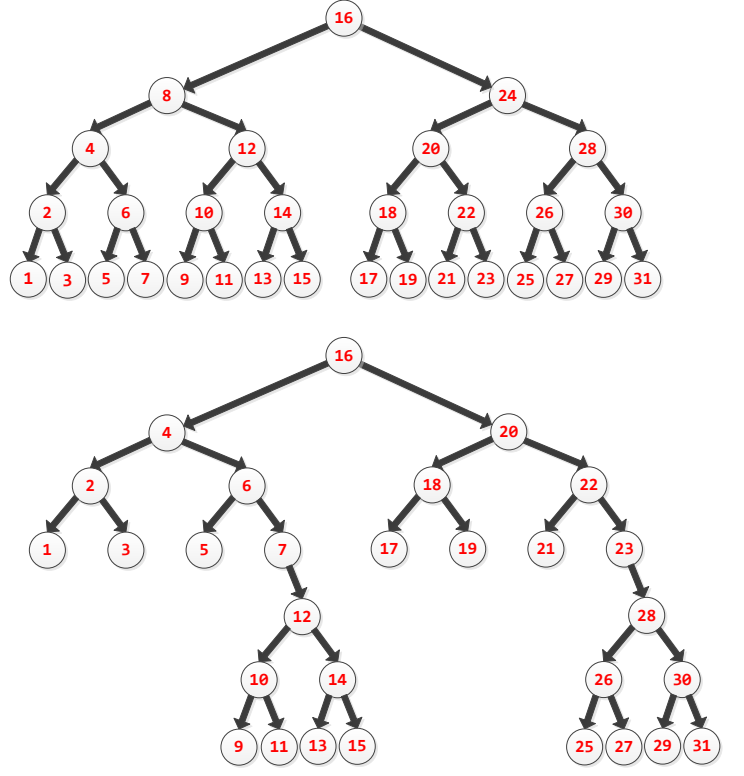
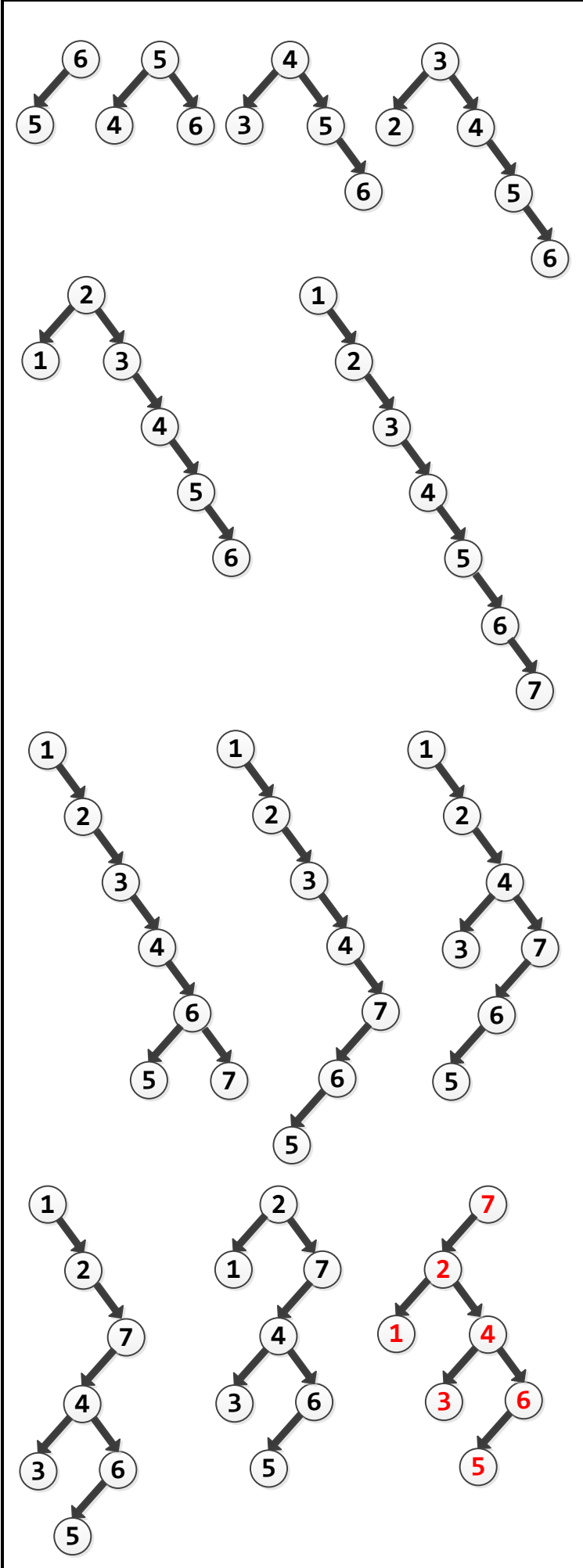
```
void print3(DoublyNode* node) // print1()
{
    if (node == header) return; //5P
    else print3(node->prev); //5P
    cout << node->elem << endl; //5P
}

void print4(DoublyNode* node) // print2()
{
    cout << node->elem << endl; //5P
    if (node->prev==header) return; //5P
    else print4(node->prev); //5P
}
```

2. Çift yönlü bağlı listeden eleman silen removeOrdered() adlı fonksiyondaki boş satırlara gerekli kodları yazınız. (10P)

6 5 4 3 2 1 7

3. Yukarıdaki verileri splay ağacına yerleştiriniz? (20P)



4. İkili ağaçtan 8 ve 24'ü yukarıdaki gibi silmek üzere aşağıdaki kodu tamamlayınız. (20P)

```
if( p->left != NULL && p->right != NULL)
{
    if(parent->left == p)
    {
        parent->left = p->left;
        p->left->par = parent;
        temp = p->left;

        while(temp->right != NULL) temp=temp->right;

        temp->right = p->right;
        temp->right->par = temp;
    }
    else
    {
        parent->right = p->left;
        p->left->par = parent;
        temp = p->left;

        while(temp->right != NULL) temp=temp->right;

        temp->right = p->right;
        temp->right->par = temp;
    }
    delete p;
}
```

5. Synonym Chaining yöntemini avantajını belirterek açıklayınız. (10P)

Synonym Chaining yönteminde çakışan kayıtlar, linear probing gibi bir çakışma çözümleme yöntemine göre relative.txt'de uygun bir yer bulunup yazıldıktan sonra bağlı listede tutulurlar. Sorgulama yapılırken çakışma olduğunda relative.txt'den aramak yerine bağlı listeden çok daha hızlı bir şekilde erişilir.