



NUMARA :	AD SOYAD :	DEĞERLENDİRME
	İMZA :	[.....]

Mühendislik Fakültesi Sınav Uygulama Yönergesi'ndeki kurallara uyulmalıdır. Sorular Bölüm Program Çıktıları'ndan 1,4,12 ile ilişkilidir.

```

void insertOrdered(SinglyNode* newNode,
                   SinglyNode* current)
{
    if(.....)
    {
        newNode->next = current->next;
        current->next = newNode;
    }
    else
        insertOrdered(newNode, current->next);
}

int main()
{
    SinglyLinkedList list; SinglyNode* newNode;

    list.head = new SinglyNode;
    list.head->score = 0;
    list.head->next = NULL;

    newNode = new SinglyNode;
    newNode->elem = "Paul"; newNode->score = 720;
    list.insertOrdered(newNode, list.head);

    newNode = new SinglyNode;
    newNode->elem = "Rose"; newNode->score = 590;
    list.insertOrdered(newNode, list.head);

    newNode = new SinglyNode;
    newNode->elem = "Anna"; newNode->score = 660;
    list.insertOrdered(newNode, list.head);

    newNode = new SinglyNode;
    newNode->elem = "Mike"; newNode->score = 1105;
    list.insertOrdered(newNode, list.head);
}

```

8 4 12 2 6 10 14 1 3 5 7 9 11 13 15

2. Yukarıdaki verilerin ikili ağaca eklendiği varsayılsın. Bu ağacın inorder, preorder ve postorder gezinme çıktıları ile yeni ağaçlar oluşturulduğunda bu ağaçların hangisi ilk ağaçla aynı olur? (25P)

Yanlış cevaptan 5P kırılacaktır.

- (A) inorder
- (B) preorder
- (C) postorder

1. **insertOrdered()** fonksiyonunu tamamlayınız. (25P)

Yanlış cevaptan 5P kırılacaktır.

- (A) if ((current == NULL)
|| (newNode->score <= current->score))
- (B) if ((current->next == NULL)
|| (newNode->score <= current->score))
- (C) if ((current == NULL)
|| (newNode->score <= current->next->score))
- (D) if ((current->next == NULL)
|| (newNode->score <= current->next->score))

```

void insertOrdered(const string& e, const int& i)
{
    DoublyNode* newNode      = new DoublyNode;
    newNode->elem          = e;
    newNode->score         = i;

    DoublyNode* current = header->next;

    while (current != trailer)
    {
        if (newNode->score >= current->score)
            current = current->next;
        else
            break;
    }

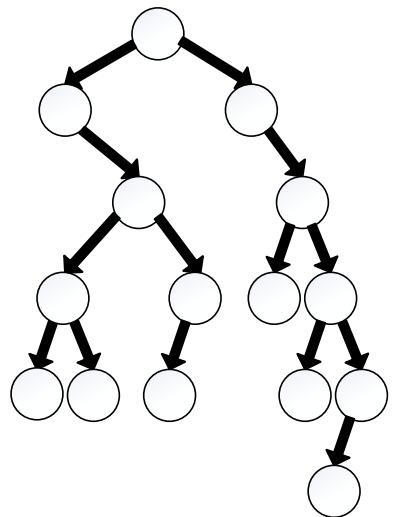
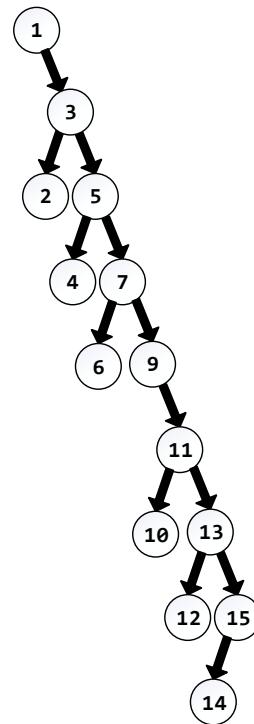
    newNode->next      = current;
    newNode->prev      = current->prev;
    .....
    .....
}

```

3. `insertOrdered()` fonksiyonundaki satırları için aşağıda verilen kodlardan hangisi listeye hatalı ekleme yapar? (25P)

Yanlış cevaptan 5P kırılacaktır.

- (A) `current->prev->next` = `newNode`;
`newNode->next->prev` = `newNode`;
- (B) `current->prev->next` = `newNode`;
`current->prev` = `newNode`;
- (C) `current->prev` = `newNode`;
`current->prev->next` = `newNode`;
- (D) `newNode->prev->next` = `newNode`;
`newNode->next->prev` = `newNode`;
- (E) `newNode->next->prev` = `newNode`;
`newNode->prev->next` = `newNode`;



4. Yukarıda soldaki Splay ağacına 8'i ekleyiniz. Ağacın son halini sağdaki boş ağaca yazınız. (25P)