



NUMARA : .....		AD SOYAD : .....		DEĞERLENDİRME	
Sınavda Uyulması Gereken Kurallar		İMZA : .....		[.....]	.....
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.					

```
DoublyLinkedList* fList(DoublyLinkedList* list1)
{
    DoublyLinkedList* list2 = new DoublyLinkedList();
    DoublyNode* nodeA = NULL;
    DoublyNode* nodeB = NULL;

    while (!list1->empty())
    {
        nodeA = list1->header->next;
        nodeB = list1->header->next->next;

        while (nodeB != list1->trailer)
        {
            if (nodeB->score < nodeA->score)
            {
                nodeA = nodeB;
                nodeB = nodeB->next;
            }
            else
                nodeB = nodeB->next;
        }

        list2->addBack(nodeA->elem, nodeA->score);
        list1->remove(nodeA);
    }

    return list2;
}

void main()
{
    DoublyLinkedList* list1 = new DoublyLinkedList();
    list1->addFront("Paul", 720);
    list1->addFront("Rose", 590);
    list1->addFront("Jack", 510);
    list1->addFront("Anna", 660);
    list1->addFront("Rob", 750);

    DoublyLinkedList* list2 = fList(list1);
    list2->printH2T();
}
```

1. fList() fonksiyonu ne iş yapar? Kısaca açıklayınız.(30P)

```
int biC(int n, int k)
{
    if (k == 0) return 1;
    if (k == n) return 1;
    return biC(n - 1, k-1) + biC(n - 1, k);
}

void main()
{
    for (int n = 0; n < 5; n++)
    {
        for (int k = 0; k <= n; k++)
        {
            cout << biC(n, k) <<" ";
        }

        cout << endl;
    }
}
```

2. Yukarıdaki programın çıktısı nedir? (30P)

```

bool empty()
{
    return (header->next == trailer);
}

void addFront(const string& e, const int& i)
{
    add(header->next, e, i);
}

void add(DoublyNode* v, string& e, int& i)
{
    DoublyNode* u    = new DoublyNode;
    u->elem           = e;
    u->score          = i;
    .....
    .....
    .....
    .....
}

void printH2T()
{
    if (empty())
    {
        cout << "List is empty !" << endl;
        return;
    }

    DoublyNode* first = header;
    while (!(first->next == trailer))
    {
        cout << first->next->elem <<
            "\t" << first->next->score << endl;
        first = first->next;
    }
}

void main()
{
    DoublyLinkedList list;
    list.addFront("Rob", 750);
    list.addFront("Paul", 720);
    list.printH2T();
}

```

3. Yukarıdaki programda add() fonksiyonunda ..... ile temsil edilen satırlar:

i) (10P)

(Yanlış cevaptan 5P kırılacaktır)

```

u->next      = v;
u->prev      = v->prev;
v->prev      = u;
v->prev->next = u;

```

olduğunda printH2T() fonksiyonu :

- (A) Liste elemanlarını yazar.
- (B) "List is empty !" yazar.
- (C) Sonsuz döngüye girer.

ii) (10P)

(Yanlış cevaptan 5P kırılacaktır)

```

u->next      = v;
v->prev->next = u;
u->prev      = v->prev;
v->prev      = u;

```

olduğunda printH2T() fonksiyonu :

- (A) Liste elemanlarını yazar.
- (B) "List is empty !" yazar.
- (C) Sonsuz döngüye girer.

iii) (10P)

(Yanlış cevaptan 5P kırılacaktır)

```

u->next      = v;
v->prev->next = u;
v->prev      = u;
u->prev      = v->prev;

```

olduğunda printH2T() fonksiyonu :

- (A) Liste elemanlarını yazar.
- (B) "List is empty !" yazar.
- (C) Sonsuz döngüye girer.

iv) (10P)

(Yanlış cevaptan 5P kırılacaktır)

```

u->next      = v;
v->prev      = u;
u->prev      = v->prev;
v->prev->next = u;

```

olduğunda printH2T() fonksiyonu :

- (A) Liste elemanlarını yazar.
- (B) "List is empty !" yazar.
- (C) Sonsuz döngüye girer.