Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

BIL 205 Veri Yapıları
Bütünleme Sınavı, 31.01.2013, 13:00, D1
Süre : **75** Dakika

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.

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

void main()
{
   DoublyLinkedList list;

   list.insertOrdered("Paul", 720);
   list.insertOrdered("Rose", 590);
   list.insertOrdered("Anna", 660);

   list.print(list.header->next);
}
```

1. **a)** Yukarıdaki programın çıktısı nedir? **(10P)**

b) `if(node==trailer)` kısmı `if(node->next==trailer)` şeklinde olsaydı çıktı ne olurdu? Sebebini açıklayınız. **(30P)**

ÇIKTI

7 5 4 3 2 1 6

2. Yukarıdaki verileri splay ağacına yerleştiriniz? **(25P)**

← Bu soruyu kağıdın arka yüzüne çözünüz

```
void removeOrdered(const string& e, const int& i)
{
  SinglyNode* current        = head;
  SinglyNode* previous       = head;

  if((current->elem == e) && (current->score == i) )
  {
     head = current->next;
     delete current;
     return;
  }

  current = current->next;
  while (current != NULL)
  {
     if((current->elem == e)&&(current->score == i))
     {
        ........................................;
        delete current;
        return;
     }

     ........................................ ;
     ........................................ ;
  }

  if(current==NULL) cout<<e<<" is not found"<<endl;
}
```

3. Tek yönlü bağlı listeden eleman silen **removeOrdered()** adlı fonksiyondaki boş satırlara gerekli kodları yazınız. **(15P)**

4. Kuyruk veri yapısına ait `enqueue()` ve `dequeue()` fonksiyonları, dairesel bağlı listenin `add()`, `advance()` ve `remove()` fonksiyonları ile gerçeklenirken `dequeue()` için `remove()` yeterli olurken `enqueue()` için neden `add()`'in peşine `advance()` işlemi de yapmak gerekir? **(20P)**