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

BIL 205 Veri Yapıları
1. Arasınav, 19.11.2013, 13:00, D-2, D-9
Süre : **90** Dakika

CEVAPLAR

```
void print1(DoublyNode* node)
{
    cout << node->elem << node->score << endl;

    if (node->next == trailer) return;

    print1(node->next);
}

void print2(DoublyNode* node)
{
    if (node == trailer) return;

    cout << node->elem << node->score << endl;

    print2(node->next);
}

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);
    list.print2(list.header->next);
}
```

1.

a) Yukarıdaki **print1()** ve **print2()** fonksiyonlarının çıktıları nelerdir? **(20P)**

print1()	print2()
Rose 590	Rose 590
Anna 660	Anna 660
Paul 720	Paul 720

b) **print1()** ve **print2()** kendilerini recursive olarak kaç kez çağırırlar? **(10P)**

print1() kendini recursive olarak **2** kez çağırır.

print2() kendini recursive olarak **3** kez çağırır.

```
void quadruple(int A[], int i, int n)
{
    if (n == 1) cout << A[i] << endl;
    else
    {
       quadruple( A, i +   n/4, n/4 );
       quadruple( A, i        , n/4 );
       quadruple( A, i + 3*n/4, n/4 );
       quadruple( A, i + 2*n/4, n/4 );
    }
}

void main()
{
  int A[16]={1,2,3,4,5,6,7,8,
     9,10,11,12,13,14,15,16};

  quadruple(A, 0, 16);
}
```

Çıktı

6
5
8
7
2
1
4
3
14
13
16
15
10
9
12
11

2. Yukarıdaki programın çıktısı nedir? **(30P)**

```
SinglyNode* SinglyLinkedList::funcA(SinglyNode* node)
{
   if(node->next == NULL) return node;
   else funcA(node->next);
}

void SinglyLinkedList::funcB(SinglyNode* node)
{
   if(node->next == NULL)
   {
      delete head;
      head = NULL;
      return;
   }

   if (node->next->next == NULL)
   {
      delete node->next;
      node->next = NULL;
      return;
   }

   funcB(node->next);
}

SinglyLinkedList* SinglyLinkedList::funcC()
{
  SinglyLinkedList* newList = new SinglyLinkedList();

  SinglyNode* node           = funcA(head);
  newList->head              = new SinglyNode();
  newList->head->elem        = node->elem;
  newList->head->score       = node->score;
  SinglyNode* tempHead       = newList->head;
  funcB(head);

  while(head != NULL)
  {
     node                    = funcA(head);
     tempHead->next          = new SinglyNode();
     tempHead->next->elem    = node->elem;
     tempHead->next->score   = node->score;
     tempHead                = tempHead->next;
     funcB(head);
  }
  tempHead->next     = NULL;

  return newList;
}

void main()
{
  SinglyLinkedList* list = new SinglyLinkedList();

  list->insertOrdered("Mike", 1105);
  list->insertOrdered("Rob",  750);
  list->insertOrdered("Paul", 720);
  list->insertOrdered("Anna", 660);
  list->insertOrdered("Rose", 590);
  list->insertOrdered("Jack", 510);

  SinglyLinkedList* newList = list->funcC();
  newList->print();

  ::getchar();
}
```

3. Yandaki programın çıktısı nedir? **(20P)**

Mike 1105
Rob 750
Paul 720
Anna 660
Rose 590
Jack 510

funcA ne iş yapar? (1 cümle ile açıklayınız) (5P)

Listenin son elemanını döndürür.

funcB ne iş yapar? (1 cümle ile açıklayınız) (5P)

Listenin son elemanını siler.

funcC ne iş yapar? (1 cümle ile açıklayınız) (10P)

Listeyi reverse yapar.