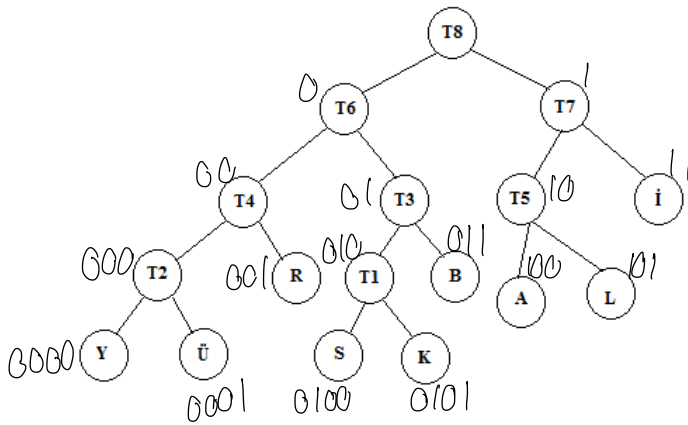


Veri Yapıları Dersi
2012-2013 GÜZ DÖNEMİ
FİNAL SINAVI

- 1) Aşağıdaki huffman kodları ağaca bakarak yazıya çeviriniz. (10 puan)



- a) 010010001011000010001101
SAKARÜL
b) 010100010010110000
KÜRBY
c) 1011101101000001100
LİBSÜA

- 2) Prototipi aşağıda verilen **findHeight** fonksiyonu bir ikili ağacın yüksekliğini özyinelemeli (recursion) olarak bulmaktadır. Bu fonksiyonun gövdesini yazınız (C++ dili kullanılacaktır) (15 puan)

```
int findHeight(Node * pRoot) {
```

```
if(pRoot==NULL) return 0;
```

```
return 1 + max(findHeight(pRoot->left), findHeight(pRoot->right));
```

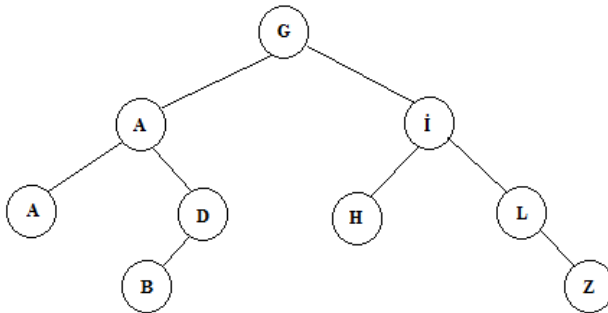
```
}
```

*if (pRoot == NULL)
return -1;*

return 1 + max (Height (pRoot->sag), Height (pRoot->sol))

Not: Bu tek çözüm olmayıp başka yollarla da çözülebilmektedir.

- 3) Aşağıdaki ağaçları preorder ve postorder olarak okuyup gerekli yere yazınız. (15 puan)

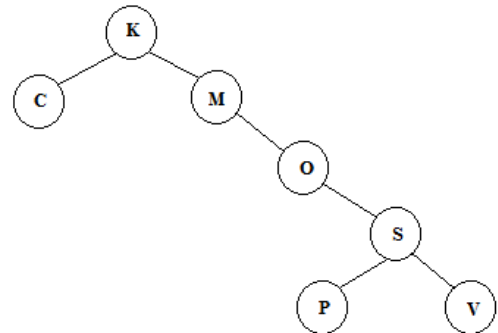


PreOrder : **GAADBİHLZ**

PostOrder: **ABDAHZLİG**

Preorder = G A A D B İ H L Z

PostOrder = A B D A H Z L İ G



PreOrder : **KCMOSP V**

PostOrder : **CPVSOMK**

Preorder = K C M O S P V

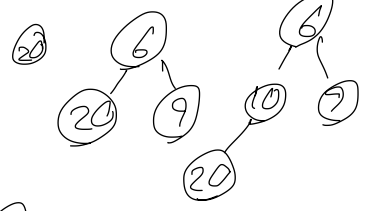
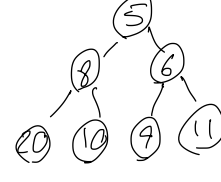
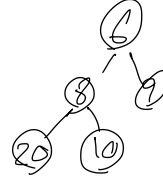
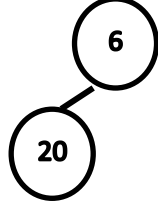
Postorder = C P V S O M K

- 4) 20,6,9,10,8,5,11 sayıları sıra ile (küçükten büyüğe doğru sıralayan) bir heap ağacına eklenmektedir. Buna göre her sayı eklendikten sonra heap ağacının ve ağacın yerleştirildiği dizinin görüntüsünü çizin. (15 puan)

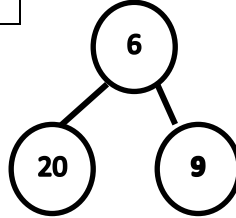
20						
----	--	--	--	--	--	--



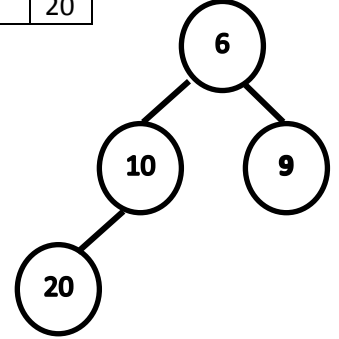
6	20					
---	----	--	--	--	--	--



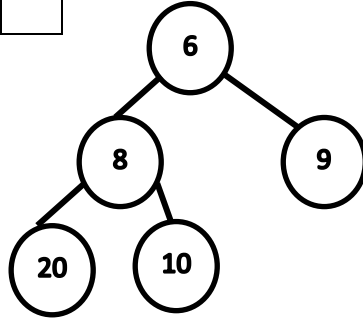
6	20	9				
---	----	---	--	--	--	--



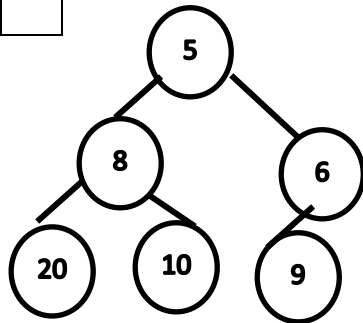
6	10	9	20			
---	----	---	----	--	--	--



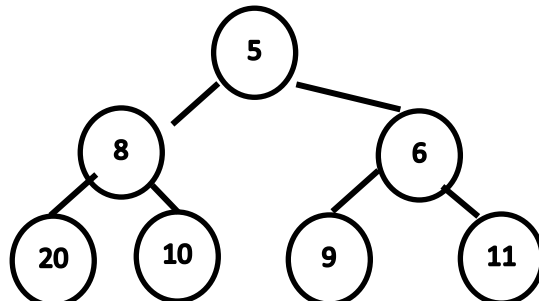
6	8	9	20	10		
---	---	---	----	----	--	--



5	8	6	20	10	9	
---	---	---	----	----	---	--

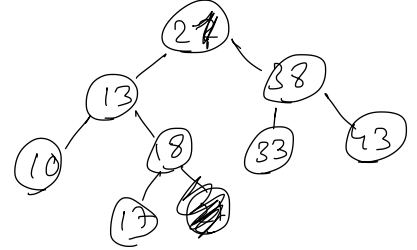
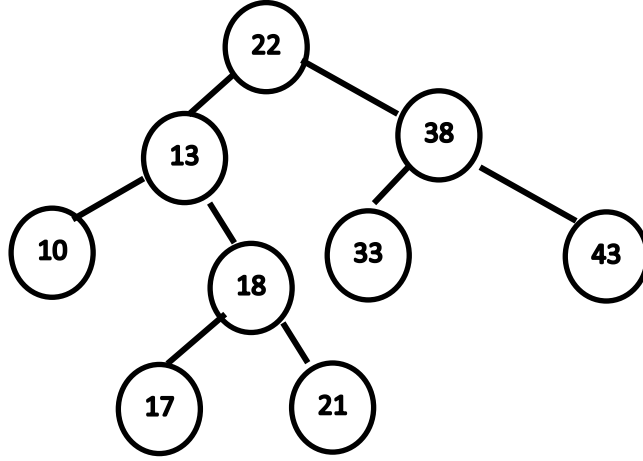


5	8	6	20	10	9	11
---	---	---	----	----	---	----

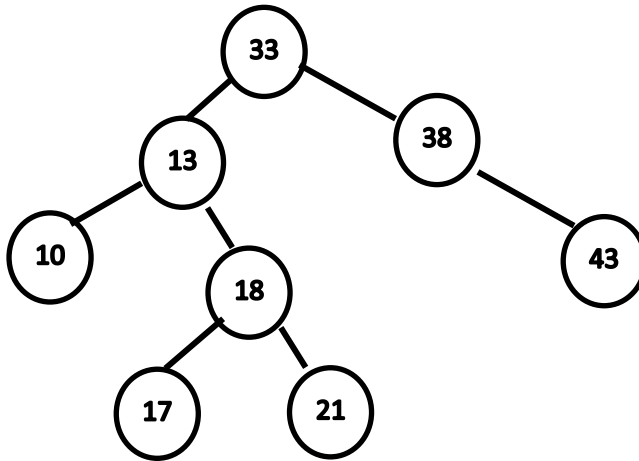


5) 22,13,18,38,43,33,17,21,10,22 sayıları sıra ile bir ikili arama ağacına yerleştirilmektedir. (15 puan)

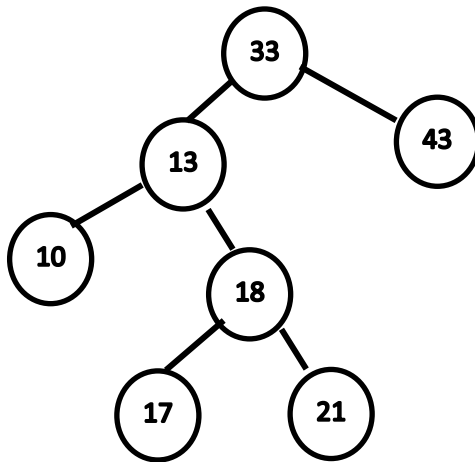
- Bu ağacı çiziniz
- Ağaçtan 22 sayısı çıkartıldığında ağacın yeni durumunu çiziniz.
- Ağaçtan 38 sayısı çıkartıldığında ağacın yeni durumunu çiziniz. (b den sonra yapılacak)



b)



c)



6) (15 puan)

```
class Kisi{
public:
    Kisi(int tc,string ad){
        TC = tc;
        Ad = ad;
    }
    int TC;
    string Ad;
};
```

```
class Hucre
{
public:
    Hucre():Durum(0){};
    int Durum;
    Kisi Veri;
};
```

```
class HashTablosu
{
public:
    Hucre HashDizisi[10];
    bool Ekle(Kisi veri);
    Kisi Cikar(int TC);
};
```

```
HashTablosu tablo;
tablo.ekle(Kisi(93,"Murat"));
tablo.ekle(Kisi(123,"Osman"));
tablo.ekle(Kisi(133,"Ahmet"));
tablo.ekle(Kisi(144,"Mehmet"));
tablo.ekle(Kisi(433,"Eser"));
tablo.ekle(Kisi(125,"Sarp"));
tablo.ekle(Kisi(128,"Emrah"));
tablo.ekle(Kisi(127,"Erbil"));
tablo.ekle(Kisi(96,"Sinan"));
tablo.ekle(Kisi(33,"Ali"));
tablo.ekle(Kisi(10,"Ayça"));
//1.DURUM
tablo.cikar(144);
tablo.cikar(93);
tablo.cikar(63);
//2.DURUM
tablo.ekle(Kisi(99,"Şükran"));
//3.DURUM
```

Yukarıda, Hash tablosu ve tablonun hücre sınıflarının prototipleri verilmiştir. Buna göre solda oluşturulan hash tablosunun görüntüsünü her bir durum için ayrı ayrı çiziniz. (işlemler art arda gerçekleşmektedir)

0	127
1	96
2	33
3	123
4	123
5	123
6	123
7	433
8	125
9	128

1.DURUM

1	127	Erbil	0
1	96	Sinan	1
1	33	Ali	2
1	93	Murat	3
1	123	Osman	4
1	133	Ahmet	5
1	144	Mehmet	6
1	433	Eser	7
1	125	Sarp	8
1	128	Emrah	9

2.DURUM

1	127	Erbil	0
1	96	Sinan	1
1	33	Ali	2
2			3
1	123	Osman	4
1	133	Ahmet	5
2			6
1	433	Eser	7
1	125	Sarp	8
1	128	Emrah	9

3.DURUM

1	127	Erbil	0
1	96	Sinan	1
1	33	Ali	2
1	99	Şükran	3
1	123	Osman	4
1	133	Ahmet	5
2			6
1	433	Eser	7
1	125	Sarp	8
1	128	Emrah	9

7) Tek yönlü bir bağlı listenin düğümlerini hafızadan kaldıran aşağıdaki fonksiyonun gövdesini yazınız.(15p)

```
void deletelist(Node* pHeader) {
```

```
    while(pHeader!=NULL)
    {
```

```
        Node* pTemp = pHeader;
```

```
        pHeader = pHeader -> pNext; (pSonraki ismi de kabul edilir.)
```

```
        delete pTemp;
```

```
    }
```

```
} while( head) {
```

```
    Node* deleted = head
```

```
    head = head -> next;
```

```
    delete deleted
```

```
}
```