

SAÜ - BİLGİSAYAR MÜH. BÖL. 2013-2014 GÜZ VERİ YAPILARI DERSİ BÜTÜNLEME SINAVI

1-) Ebeveyn Düğüm: En az bir çocuğu olan düğüme denir. Dizi ile gerçekleştirilmemiş ikili arama ağacında ebeveyn düğüm sayısını bulan fonksiyonu yazınız. (15 p)

```
int EbeveynDugumSayisi(){
    return EbeveynDugumSayisi(root);
}
int EbeveynDugumSayisi(Dugum<Nesne> *alt_root)
{
    if(alt_root == NULL) return 0;
    if(alt_root->sol != NULL || alt_root->sag != NULL){
        return 1 + EbeveynDugumSayisi(alt_root->sol) + EbeveynDugumSayisi(alt_root->sag);
    }
    else return 0;
}
```

2.) Aşağıdaki kod derlenip çalıştırıldığında cout mesajları ekrana ne yazacaktır? Karşısına yazınız. (15 p)

```
int main(){
    int x = 5, y = 3;
    int *Z = new int[2], **R = &Z;
    Z[0] = x;
    (*Z) -= 2;
    if (*Z == x)x++;
    if (*Z == **R)Z[1]=10;
    (**R)--;
```

0x28ff1c	x
0x28ff18	y
0x28ff10	R
0x28ff06	Z

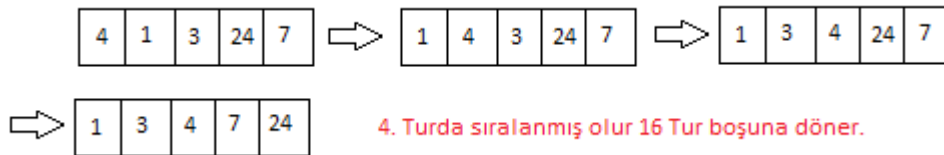
cout << x << endl;	5
cout << &Z << endl;	0x28ff06
cout << *Z << endl;	2
cout << R << endl;	0x28ff06
cout << **R << endl;	2

```
return 0;
```

```
}
```

3.) Aşağıdaki diziyi Bubble (Kabarcık) Sort kullanarak sıralayınız. Sıralama işleminin her adımı çizilerek gösterilmelidir. (10p)

4	1	3	24	7
---	---	---	----	---



4. Turda sıralanmış olur 16 Tur boşuna döner.

Ad/Soyad:

Numara:

4.)

```
class queue
{
public:
    queue();
    void push(int yeni);
    int pop();
    bool bosmu();//boş ise true döner
private:
    int veri[10];
    int KB;
    int KS;
};
```

sira adlı kuyruğun Başlangıç durumu

0	1	2	3	4	5	6	7	8	9
5	6	7	8	10					

KB

KS

Prototipi sağda verilmiş olan dairesel kuyruk sınıfından oluşturulan **sira** nesnesine aşağıdaki işlemler uygulanmaktadır. Buna göre **sira** isimli kuyruğun son durumunu çiziniz. (Kuyruk boyutu sabittir) (12 p)

```
int ch;
for(int i=0;i<10;i++)
{
    if((ch=sira.pop())%2)
        sira.push(ch+1);
}
while(ch--)
{
    sira.push(ch/3);
}
```

0	1	2	3	4	5	6	7	8	9
0	6	7	8	10	6	8	1	2	0
KS									KB

5.)

Dikdörtgen içerisinde bulunan ve ikili arama ağacına düğüm eklemek üzere kullanılan kod parçasındaki boşlukları (toplam 6 adet) uygun bir şekilde doldurunuz. (Sınıf içerisindeki üye değişkenlere erişirken uygun olan set ve get fonksiyonlarını kullanınız. Üyelere doğrudan erişmeyiniz.) (18 p)

```
typedef double DugumVeriTipi;
```

```
class Dugum
```

```
{
```

```
private:
```

```
    DugumVeriTipi dugumVerisi;
```

```
    Dugum* solDugum;
```

```
    Dugum* sagDugum;
```

```
public:
```

```
    void setDugumVerisi(DugumVeriTipi veri);
```

```
    DugumVeriTipi getDugumVerisi();
```

```
    void setSolDugum(Dugum *veri);
```

```
    Dugum *getSolDugum();
```

```
    void setSagDugum(Dugum *veri);
```

```
    Dugum *getSagDugum();
```

```
    void dugumEkle(Dugum* yeniDugum);
```

```
    void dugumYazdirInOrder() const;
```

```
    void dugumYazdirPostOrder() const;
```

```
    void dugumYazdirPreOrder() const; //Artan
```

```
    void dugumYazdirAzalan() const; //Azalan
```

```
    bool ara(DugumVeriTipi value) const;
```

```
};
```

```
void Dugum::dugumEkle(Dugum* yeniDugum)
```

```
{
```

```
    if(yeniDugum->getDugumVerisi() == this->getDugumVerisi())
```

```
    {
```

```
        cout<<"Eklemeye çalıştığınız veri mevcuttur";
```

```
        return;
```

```
    }
```

```
    if (yeniDugum->getDugumVerisi() < this->getDugumVerisi())
```

```
    {
```

```
        if (this->getSolDugum() == NULL) this->setSolDugum(yeniDugum);
```

```
        else solDugum->dugumEkle(yeniDugum);
```

```
    }
```

```
    else if (this->getDugumVerisi() < yeniDugum->getDugumVerisi())
```

```
    {
```

```
        if (this->getSagDugum() == NULL) this->setSagDugum(yeniDugum);
```

```
        else sagDugum->dugumEkle(yeniDugum);
```

```
    }
```

```
}
```

Ad/Soyad:

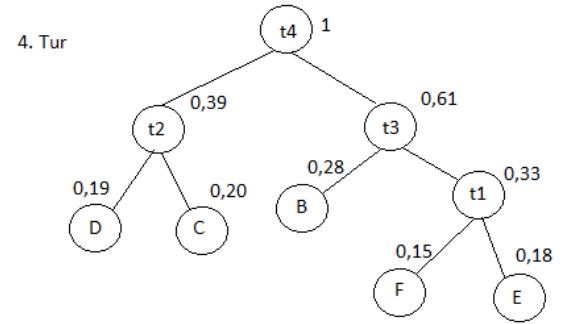
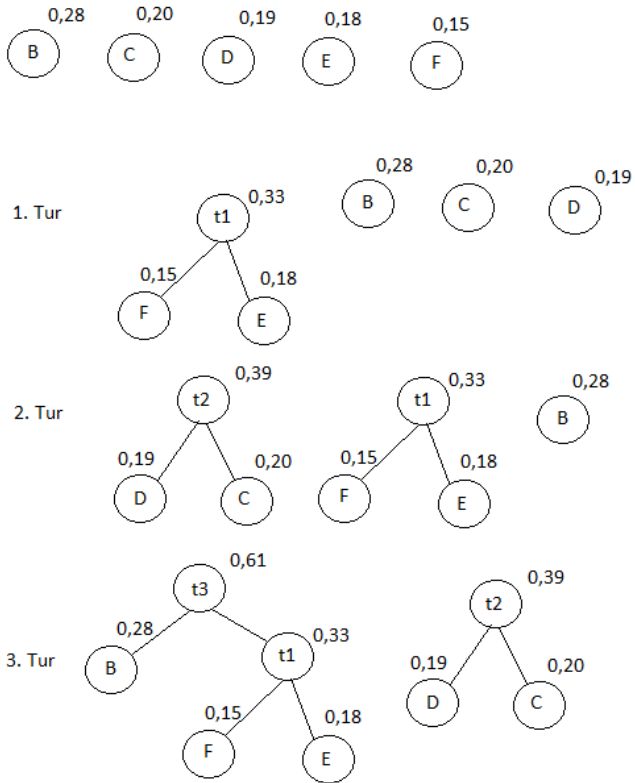
Numara:

6.) Aşağıdaki Hash fonksiyonu string değerinin hangi indekste tutulacağını hesaplayan bir fonksiyondur. Fonksiyonun gövdesini yazınız. (15 p)

```
int Hash(string deger, int diziUzunluk){
    int asciiDeger = 0;
    for (int i = 0; i < deger.length(); i++)
    {
        asciiDeger += int(deger.at(i));
    }
    return asciiDeger % diziUzunluk;
}
```

7) Aşağıda bir yazıda karakterlerin **dağılım olasılıkları** verilmiştir. Buna göre Huffman ağacını çizin ve her karakterin yayına kodunu yazınız. (15 p)

Karakter	Olasılık	Huffman Kodu
B	0,28	10
C	0,20	01
D	0,19	00
E	0,18	111
F	0,15	110



Huffman Ağacı

