Ad/Soyad: 15.01.2014
Numara: Süre: 80 dakika

SAÜ BİLG. VE BİLİŞ. BİL. FAK. BİLGİSAYAR MÜH. BÖL. 2013-2014 GÜZ VERİ YAPILARI DERSİ FİNAL SINAVI

1-) **Dizi ile gerçekleştirilmemiş** İkili arama ağacında düğüm sayısını bulan fonksiyonu yazınız. (15 p)

```
int BinaryTree::Uzunluk(Node *alt_root) const{
        if(alt_root == NULL) return 0;
        else return (Uzunluk(alt_root->sol) + 1 + Uzunluk(alt_root->sag));
}
int BinaryTree::Uzunluk() const{
        return Uzunluk(root);
}
```

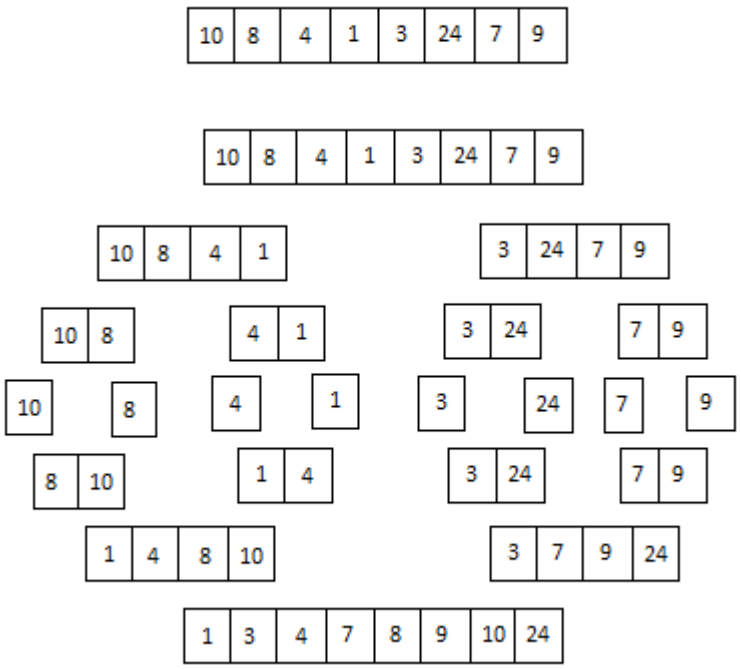
2.) Aşağıdaki kod derlenip çalıştırıldığında cout mesajları ekrana ne yazacaktır? Karşısına yazınız. (10 p)

```
int main(){
        int x = 14, y = 53;
        int *P = &x, **R = &P;
        (*P) -= 4;
        P = new int(10);

        if (*P == x)y += 2;
        if (*P == **R)y = 5;
        (**R)++;

        cout << x << endl;
        cout << &P << endl;
        cout << *P << endl;
        cout << R << endl;
        cout << **R << endl;

        return 0;
}
```

0x28ff1c	x
0x28ff18	y
0x28ff14	P
0x28ff10	R

cout << x << endl;	10
cout << &P << endl;	0x28ff14
cout << *P << endl;	11
cout << R << endl;	0x28ff14
cout << **R << endl;	11

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



Önemli: Her sayfada mutlaka adınız yazmalıdır. Her soruyu kendi sayfasına çözünüz sığmaz ise aynı sayfanın arkasını kullanınız.

Ad/Soyad:
Numara:

4.) Öz yinelemeli fonksiyonlar yığın kullanılarak döngüsel hale getirilebilirler. Örneğin faktoriyel hesabı yapan ***f*** fonksiyonunun döngüsel hali ***f_stack*** fonksiyonudur. Buna göre fibonacci sayılarını hesaplayan ***Fib*** fonksiyonunun döngüsel fonksiyon (Stack kullanan) halini yazınız. (15 p)

```
int f(int sayi) {
        if(sayi==1) return 1;
        return sayi*f(sayi-1);
}
```

```
int f_stack(int sayi) {
        Stack<int> s;
        while(sayi) s.push(sayi--);
        int sonuc =1;
        while(!s.bosmu(){
                sonuc*=s.pop();
        }
        return sonuc;
}
```

```
int Fib(int sayi)
{
        stack yigin;
        if(sayi==0)
                return 0;

        yigin.push(0);
        yigin.push(1);

        while((--sayi)>0)
        {
                int pop1 = yigin.pop();
                int pop2 = yigin.pop();
                yigin.push(pop1);
                yigin.push(pop1+pop2);
        }
        return yigin.pop();
}
```

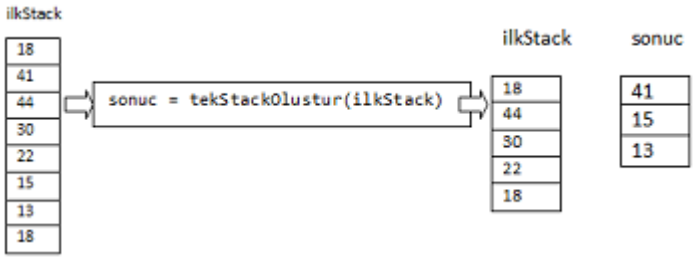
5.)

```
class stack {
public:
   stack();
   void push(int yeni);
   int pop();
   bool bosmu();//boş ise true döner
private:
       int*  veri;
       int   tepe;
};
```

Prototipi aşağıda verilen **tekStackOlustur** fonksiyonunun görevi parametre olarak aldığı **stack** nesnesindeki tek sayıları kullanarak yeni bir **stack** oluşturmaktır. Oluşturulan **stack** adresi geri döndürülmektedir. Aşağıda fonksiyonun kullanımına örnek verilmiştir. Bütün stack nesneleri heap bölgesinde oluşturulacaktır. Fonksiyonu yazınız

(10 p)

```
stack* tekStackOlustur(stack* yigin)
{
        stack* yedek = new stack;
        stack* sonuc = new stack;
        while(yigin->bosmu())
        {
                int okunan = yigin->pop();
                if(okunan%2==1)
                {
                        sonuc->push(okunan);
                }
                else
                {
                        yedek->push(okunan);
                }
        }
```



Önemli: Her sayfada mutlaka adınız yazmalıdır. Her soruyu kendi sayfasına çözünüz sığmaz ise aynı sayfanın arkasını kullanınız.

Ad/Soyad:
Numara:

```
        while(yedek->bosmu())
        {
                int okunan = yedek->pop();
                yigin->push(okunan);
        }
        delete yedek;
}
```

6.) Dikdörtgen içerisinde bulunan kod parçasındaki boşlukları (toplam 5 adet) uygun bir şekilde doldurunuz. (10 p)

```
const int eLEMANsAYISI=10;//Eleman sayısı

class HashTablosu
{
    private:
        ListeYeni* depolamaBirimi[eLEMANsAYISI];
    public:
        HashTablosu();
        int hashKoduUret(const string &giris);
        void ekle(const string &giris);
        void yazdir(const string &giris);
        void tumunuYazdir();
};
```

```
void HashTablosu::.....(const string &giris)
{

    ..............[.............(giris)%.........]->basaEkle(........);
}
```

```
void HashTablosu::ekle(const string &giris)
{

   depolamaBirimi[hashKoduUret(giris)%eLEMANsAYISI]->basaEkle(giris);
}
```

7.) a) Dengeli bir ikili arama ağacında yapılacak arama işleminin zaman karmaşıklığı Büyük-O (Big-O) gösteriminde nedir? (5 p)

O(logn)

b) **76 93 40 47 10 3 17 66 5**

Sayı dizisini 10 uzunluğundaki Hash tablosuna yerleştiriniz. Üç farklı çarpışma giderici yöntem ayrı ayrı kullanılıp ayrı Hash tabloları çizilmelidir. (Çarpışma Giderme: Zincirleme, Açık Adresleme, İkili Hashing) (10 p)

Zincirleme

0	→ 10 → 40
1	
2	
3	→ 3 → 93
4	
5	→ 5
6	→ 66 → 76
7	→ 17 → 47
8	
9	

Açık Adresleme

0	40
1	10
2	
3	93
4	3
5	5
6	76
7	47
8	17
9	66

İkili Hashing

R=7

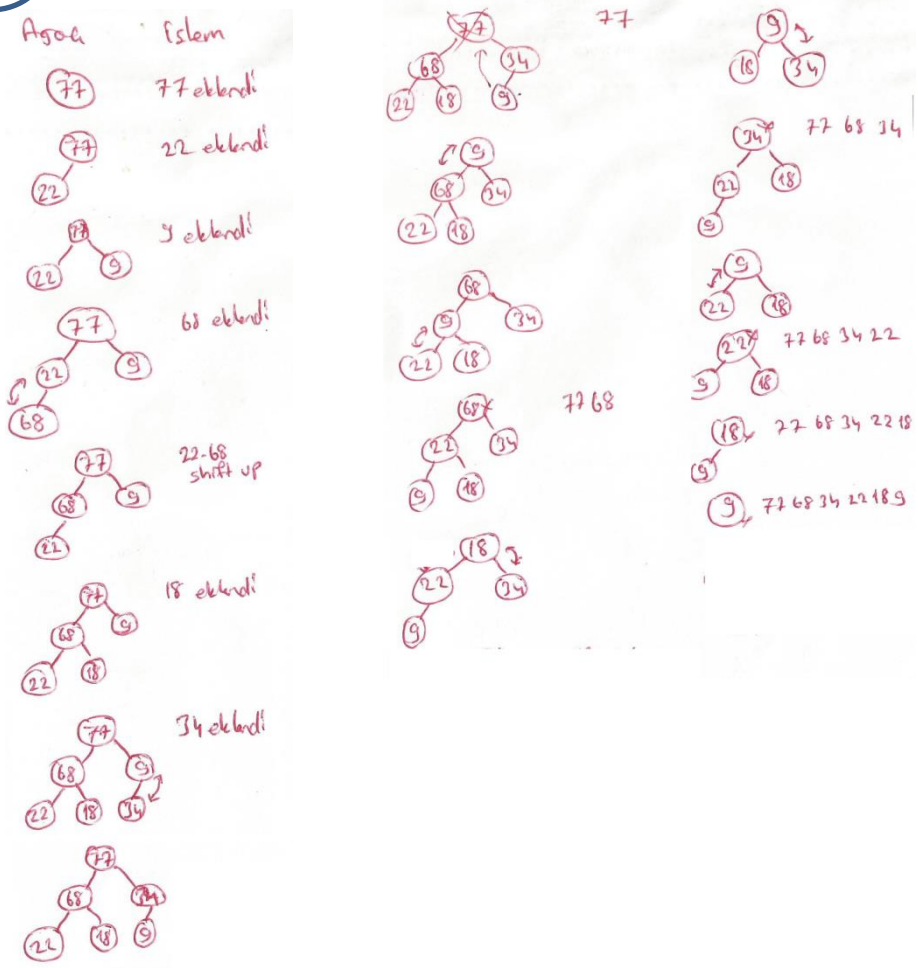
0	40
1	3
2	
3	93
4	10
5	17
6	76
7	47
8	66
9	5

Önemli: Her sayfada mutlaka adınız yazmalıdır. Her soruyu kendi sayfasına çözünüz sığmaz ise aynı sayfanın arkasını kullanınız.

Ad/Soyad:
Numara:

8.) a. 77, 22, 9, 68, 18, 34 sayılarıyla maksimum heap ağacı oluşturunuz. Her aşamada yapılan işlemler ayrı ayrı gösterilecektir.
b. Oluşturduğunuz bu ağacı kullanarak verilen sayıları büyükten küçüğe sıralayınız.
NOT: HER ADIMDA OLUŞAN AĞAÇ VE YAPILAN İŞLEMİ YAZINIZ. DİREK SONUÇ YAZILDIĞINDA PUAN VERİLMEYECEKTİR. LIST HEAP YAPISI <u>KULLANILMAMALIDIR</u> (10 p)

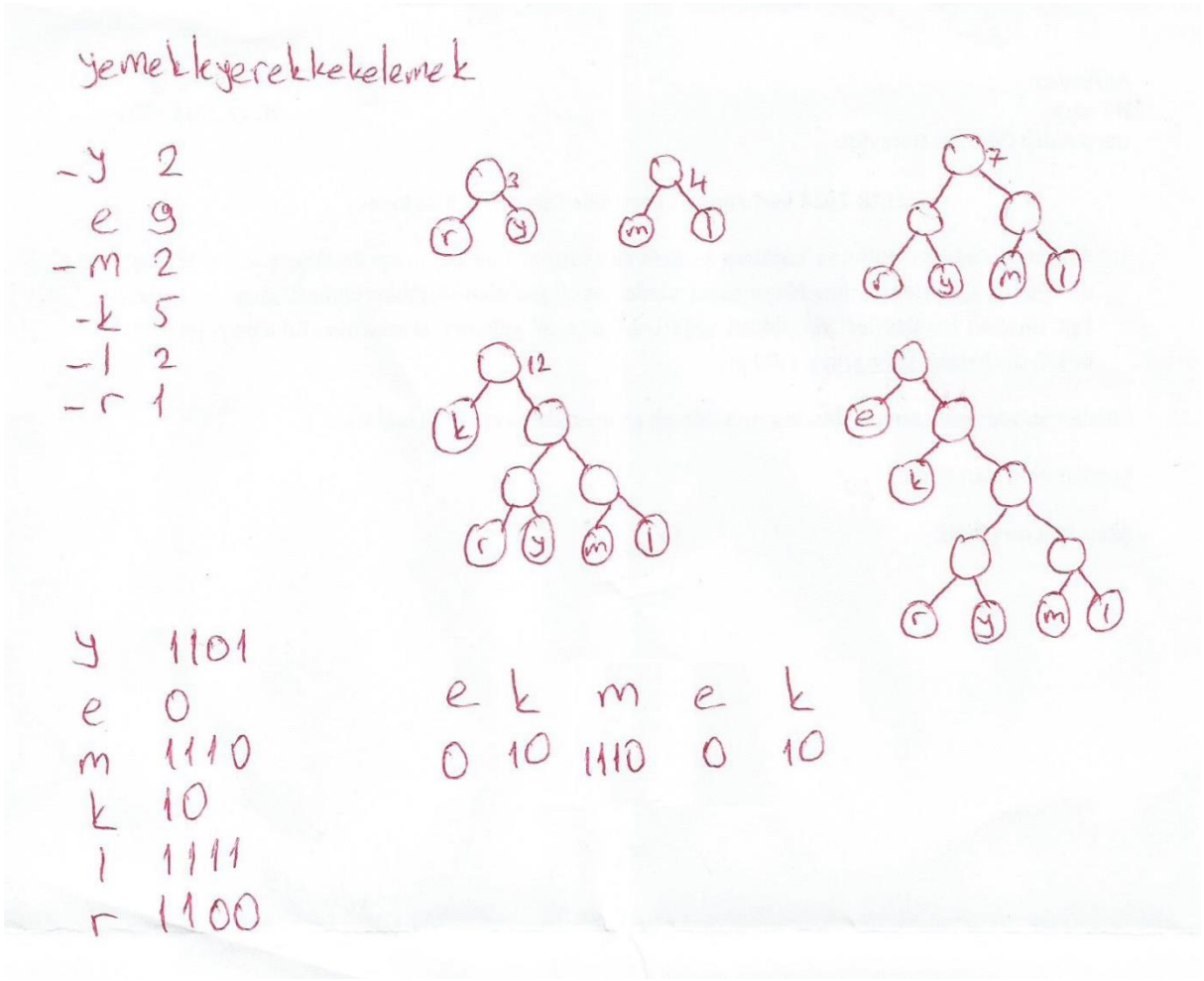
AĞAÇ	**İŞLEM**	**AĞAÇ**	**SIRALAMA LİSTESİ**
77	77 eklendi		



9.) a) **yemekleyerekkekelemek** kelimesini Huffman kodlama kullanarak sıkıştırınız. Düğümlerden ağaç oluşturulurken karakter sayısı küçük veya eşit olanlar solda, büyük olanlar ise sağda yer alacak şekilde ağacınızı oluşturunuz. Eşit sayıdaki karakterler için soldan sağa okuyunca ilk gelen harflere öncelik verilmesi gerekmektedir.
b)Oluşacak ağacınıza göre her bir harfin kodunu yazınız.
c) **ekmek** kelimesinin kodlanmış halini yazınız.
NOT: Çizmiş olduğunuz ağaç doğru olmadıktan sonra yazılan ve doğru olan Huffman koduna puan verilmez. (15 p)

Önemli: Her sayfada mutlaka adınız yazmalıdır. Her soruyu kendi sayfasına çözünüz sığmaz ise aynı sayfanın arkasını kullanınız.

Ad/Soyad:
Numara:



Önemli: Her sayfada mutlaka adınız yazmalıdır. Her soruyu kendi sayfasına çözünüz sığmaz ise aynı sayfanın arkasını kullanınız.