

NACİ SEFA GÜRSOY

## VERİ YAPILARI YAZ OKULU ÖDEVİ

Yazdığım program C++konsol uygulaması olup çalıştırıldığında ilk olarak Kisiler.txt isimli dosyayı okumaktadır.Bu dosyada her satırda bir kişi verilmiştir. Kişi bilgileri # karakteri ile ayrılmıştır.Kişi bilgileri İsim#doğumyılı#kilo şeklindedir.Uygulama kişileri satır satır dosyadan okuyup işleme sokmaktadır.

```
string ad,yas,klo;
ifstream dosya ("kisiler.txt");

if(dosya.is_open()) {

    string line;
    int sayl;

    while (!dosya.eof ()) {

        getline (dosya,ad,'#');
        getline (dosya,yas,'#');
        getline (dosya,klo,'\n');

    }
}
```

Burada yazılılan getline fonksiyonu ile bilgiler kayıba uğramadan satır satır dosyanın sonuna kadar okunuyor.getline fonksiyonunu kullanma amacım hem satır satır işlem yapmak hemde # ile ayrılmış bilgileri düzenli bir şekilde okumak.

```
sayl = stoi(yas);
toplam=2020-sayl;
```

Devamın da string olarak aldığım yas değişkeninin yani listedeki doğum yılı değişkenini ilk olarak int tipine çevirip daha sonra bunu bir değişkene atıyorum.

```
r=avl.insert(r,toplam,ad,klo);
```

Bu aldığım bilgileri insert fonksiyonu ile avl ağaçına aktarıyorum.insert fonksiyonunu açacak olursak kısaca.

```
'  
avl *avl_tree::insert(avl *r, int v,string ad,string klo) {  
    if (r == NULL) {  
        r = new avl;  
        r->d = v;  
        r->ad=ad;  
        r->klo=klo;  
        r->l = NULL;  
        r->r = NULL;  
        return r;  
    } else if (v< r->d) {  
        r->l = insert(r->l, v,ad,klo);  
        r = balance(r);  
    } else if (v >= r->d) {  
        r->r = insert(r->r, v,ad,klo);  
        r = balance(r);  
    } return r;  
}
```

Alınan bilgileri belirli bir düzenleme şekliyle düzenleyip eğer bir dengesizlik varsa her eleman eklemeden önce bu düzensizliği düzeltip yani döndürme işlemi yapıp öyle ağacı düzenliyor.Yani bir kayıp olmadan ağaca ekleniyor ve düzensizlik oluşmuyor.

```
    avl.postorder(r);  
  
    dosya.close();
```

En son olarak tüm ağacı postorder şeklinde okuyup ekrana yazdırıyor.Postorder fonksiyonunu açak olursak.

```
void avl_tree::postorder(avl *t) {  
    if (t == NULL)  
        return;  
    postorder(t->l);  
    postorder(t->r);  
  
    cout << t->ad<<"<<2020-t->d<<"<<t->klo<<endl;  
}
```

Uygulamam bu şekildeydi incelediğiniz için teşekkür ederim.