

# ÖĞRENCİ KAYIT SIRALAMA

**HÜSEYİN FIRAT ALBAYRAK-160202024**

**frat57571@hotmail.com**

## ÖZET

Bu programı bağlı liste kullanımıyla gerçekleştirdik. [3] Öncelikle öğrencileri öğrenci\_kayit.txt dosyasından aldık.[1]Sonra fonksiyonlara göndererek öğrencilere kayıt kontrolünü,numara atamasını,numara sıralamasını,öğrencilerin sınıflara dağılımını ve bonus olarak öğrenci ekleme fonksiyonlarını yazdık.

Kullandığımız

**dosyaYazdir**

**olustur**

**kayitKontrol**

**numaraAtama**

**numaraSiralama**

**enAzDagilim**

**esitDagilim,**

**ogrenciEkle**

fonksiyonlarıyla işlemlerimizi gerçekleştirdik. Yapılan işlemleri rahatça gözlemlemek için neredeyse her fonksiyonun sonuçlarını ayrı txt dosyalarına yazdırdık.

## GİRİŞ

Çalışmanın konusu programlama dersinde gördüğümüz struct yapıları ve dosya işlemleri konularını kullanarak bir sınıf planlama sistemi oluşturduk. Programda istenen bir üniversitedeki öğrencilerin kayıt edilip öğrencilere öğrenci numarası atanarak öğretim türlerine göre sınıflara en az dağılım ve eşit dağılım şeklinde sınıflara dağıtılmasıdır.

Bizim projede yapmaya çalıştığımız işler üzerine daha önceden yapılmış hazır kodlar bulamadık ama bazı hazır fonksiyonları kodumuzda kullandık.

Yaptığımız projenin amacı Bir üniversitede o dönem yeni gelen öğrencilerin üniversiteye kayıt yaptırdıktan sonra öğrencilere öğretim türleri ve giriş yaptıkları yıla göre numara verilmektedir.Sonra bu numaralar sıralanmaktadır.İsmi birden fazla geçen öğrencilerin kayıt kontrolü yapıp isimleri silindikten sonra okuldaki sınıf kapasitesine göre öğrencilerin sınıflara eşit dağılımını

**GÖRKEM GÜLMEZ-160202076**

**gorkemgulmez7@gmail.com**

yada en az sınıf dağılımı şeklinde 2 fonksiyonla öğrencileri sınıflara sınıf kapasitelerini aşmadan dağıtıyoruz.Üniversiteye sonradan katılan öğrencilerin öğrenci kaydını yapıp yeni bir öğrenci numarası veriyoruz.Programımız kısacası üniversite kayıt yerleştirme programı olarak kullanmayı amaçladık.

## TEMEL BİLGİLER

Kodumuzun öğrenci ekle,numara sıralama,numara atama,kayıt kontrol,öğrencilerin txt den alınması,dosya yazdırma işlemleri code blocks uygulamasıyla derlenip çalıştırılmıştır.

Sublime text de kodumuzu yazıp code blocks uygulamasıyla derledik.

Diğer derleyici hata verince Dev-C++ uygulamasında kodu derlemek için kullandık.

Google, Yandex gibi arama motorlarını yardım almak amacıyla kullandık.

Üniversite bölümümüzün sitesine koyulmuş olan slaytlara bakılarak dosyalama işlemlerine göz atıldı.[6][7][8]

## SONUÇLAR

Çalışmamızın sonucunda yaklaşık 450 satırlık bir kod dosyası elde ettik. Yazdığımız her bir kısmı ayrı fonksiyonlara yazarak kodu daha kolay okunabilir ve değiştirilebilir hale getirdik.

Programda bizim fark edebildiğimiz sadece bir hata bulunmaktadır. Ayrıca bonus kısım olan ogrenciEkle fonksiyonu da yazılmıştır. Bunun dışında program kısa ve hızlı olabilir.

Ogrenci struct ı içindeki char değişkenlerini sabit tutulmuştur. Bu dinamik olarak ayarlanabilseydi hafızadan kazanç sağlanabilirdi.

dosyaYazdir adlı ayrı bir fonksiyon açarak koddaki neredeyse bütün yazdırma işlerini bir fonksiyonda hallettik. Bu satır sayısını azaltmakla birlikte uğraşmamız gereken hataları azalttı.

numaraSiralama fonksiyonunda bulunan kodu verdiği hatalardan ötürü çıkarmıştık. Eğer ilk yazdığımız kodu kullanabilseydik program daha hızlı ve verimli çalışabilirdi.

esitDagilim fonksiyonu da daha az satir kodla yazılabilirdi. Ayrıca ikinci öğretim öğrencilerini düzgün

enAzDagilim fonksiyonu daha az satir kodla yazılabilirdi.

yazdıramıyor. Bunun sebebinin derleyici hatası olabileceğini düşünüyoruz.

## Diğer Bölümler

### -Bağlı Liste Oluşturma

Oğrenciler adlı bir struct oluşturarak içine gerekli değişkenleri ekledik. Öğrenciler\* olustur adlı fonksiyonda ise bu struct ın elemanlarını tek yönlü bağlı liste olarak oluşturuyoruz. Bunu da bir döngü içinde bütün öğrencileri okurken yapıyoruz. Başlangıçta sadece root (kök) adında pointer bir eleman tanımlanıyor. Daha sonrasında her öğrenci için ayrı bir struct pointer açıp birbirine pointerlar aracılığı ile bağlıyoruz. Fonksiyonun sonunda ise root değerini main e döndürüyoruz. Böylece main de çağrılan diğer fonksiyonlarda root referans değerini rahatça parametre olarak yollayabiliyoruz.

### -Öğrenci Okuma

İlk satır gereksiz olduğu için burayı atlıyoruz. Bundan sonra öğrenci bilgilerini hazır fonksiyon ile çekiyoruz[2]. Eğer öğrencinin numarası belirtilmemişse yani “-“ ise öğrencinin numarasını -1 olarak değiştiriyoruz.

### -Kayıt Kontrol

Bu fonksiyonda ismi birden fazla olan öğrenciler siliniyor ve listenin yeni hali kayıtkontrol.txt de yazdırılıyor. Programı hızlandırmak için eklenen özellikler ile öğrenci listesi isim sırasına göre yazdırıldıysa doğru sonuçlar vermektedir.

### -Numara Atama

Fonksiyonda öğrenci numarası olmayan öğrencilere numara atanmaktadır. Yılı 1700000 öğretimi 1000 ya da 2000 olarak ayarladıktan sonra yıl öğretim ve kayıt numaraları topluyoruz. Yani 17 girişli birinci öğretim 389 kayıt nolu öğrencinin okul numarası şu şekilde hesaplanıyor:  $1700000+1000+389= 1701389$ .

### -En Az Dağılım

Bu fonksiyonda öğrencileri kullanıcıdan aldığımız sınıf sayısına göre sınıf kapasitesi büyük olandan küçük olana doğru sınıf kapasitelerini doldurarak 1.öğretim,2.öğretim şeklinde sınıf sayısı kadar txt açtırarak öğrencileri txt dosyasına yazdırdık.

### -Eşit Dağılım

Bu fonksiyonda öğrencileri kullanıcıdan aldığımız sınıf sayısına göre sınıflara eşit bir şekilde dağıtıyoruz ve 1.öğretim 2.öğretim şeklinde ayırarak sınıf sayısı kadar txt dosyası açarak öğrencileri bu txt dosyalarına yazdırıyoruz.

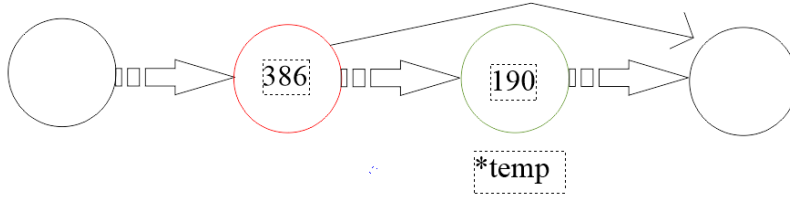
### -Öğrenci Ekle

Bu fonksiyonda kullanıcıya öğrenci eklemek isteyip istemediği soruluyor. Eğer eklemek isterse öğrencinin adı,soyadı ve öğretim türünü yine kullanıcıdan istiyor. Öğrenciyi listenin sonuna ekleyip öğrenciEkle.txt dosyasında listenin en güncel hali yazdırılıyor.

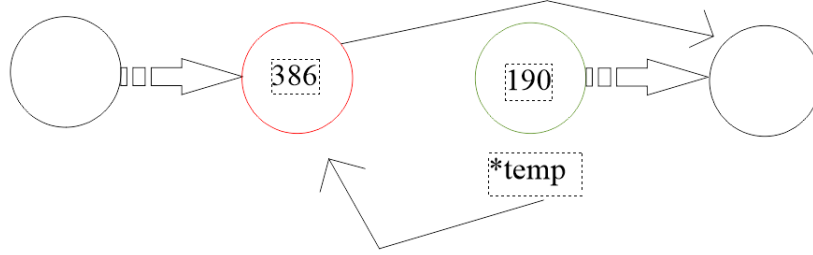
### -Numara Sıralama

Bu fonksiyonda öğrenci numaralarını küçükten büyüğe sıralayarak numaraSiralama.txt dosyasına yazdırıyoruz. Sıralama fonksiyonunda bir döngü içinde listenin sıralı iki elemanı birbirleriyle karşılaştırıyor. Eğer büyük eleman ikinci sırada ise birinci sıradaki ile yer değiştiriyor. Bu işlemi liste elemanı kadar tekrarlıyoruz.

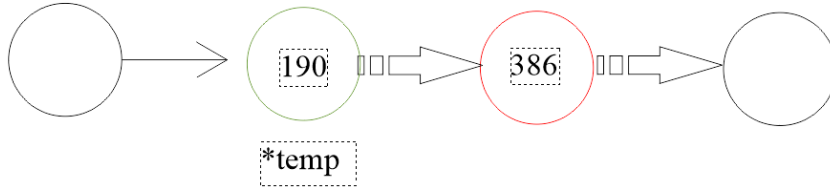
Sıralamanın mantığı aşağıdaki görsellerle de anlatılmıştır.



Şekil 1.1(Bağlı Liste)



Şekil 1.2



Şekil 1.3

```

C:\Users\Gorkem\Desktop\Test\bin\Debug\Test.exe
Rose Hart
Ryan Wright
Sophie Lyman
Una Hodges
Una Hodges
Sınıf sayısını giriniz: 3
Sınıf kapasitelerini giriniz:
1:90
2:86
3:12
Dersi alan öğrenci sayısı: 1.öğretim: 200, 2.öğretim: 199 dir.
Öğrenci eklemek ister misiniz[0,1](Varsayılan 1):
1
Adı: Gorkem
Soyadı: Gulmez
Öğretimi[1,2]-(Varsayılan 2): 2
Sınıf sayısını giriniz: 3
Sınıf kapasitelerini giriniz:
1:90
2:86
3:12
Dersi alan öğrenci sayısı: 1.öğretim: 200, 2.öğretim: 200 dir.
Öğrenci eklemek ister misiniz[0,1](Varsayılan 1):
0
En güncel dosyanın adı "final.txt"dir
Gecen süre 9.257 saniye
Process returned 0 (0x0)   execution time : 30.070 s
Press any key to continue.
  
```

Şekil 2.1

(Programın Çalışması)

1702381	Anna	Hunter	381	II
1702382	Brian	Ross	382	II
1702383	Sam	Hill	383	II
1702384	Virginia	Fisher	384	II
1702385	Jane	Pullman	385	II
1702386	Jan	Allan	386	II
1702387	Harry	Berry	387	II
1702388	Richard	Mitchell	388	II
1702389	Simon	Kelly	389	II
1702390	Elizabeth	Quinn	390	II
1702391	Gorkem	Gulmez	391	II

Şekil 2.2

## KAYNAKÇA

[https://www.tutorialspoint.com/cprogramming/c\\_file\\_io.htm](https://www.tutorialspoint.com/cprogramming/c_file_io.htm)[1]

<https://stackoverflow.com/questions/24868225/c-scanf-int-into-struct>[2]

<https://www.youtube.com/watch?v=r3uOBb3BM-0>[3]

[https://www.youtube.com/watch?v=DGi\\_gSfYfKo&t=1117s](https://www.youtube.com/watch?v=DGi_gSfYfKo&t=1117s)[4]

<https://stackoverflow.com/questions/10908668/how-do-you-read-tab-delimited-strings-from-a-txt-file-and-put-them-into-variable>[5]

[https://docs.google.com/presentation/d/1-PgaZBb0Ol4FeK80SZQzfrE5FxyHMWhLLaiyXO6g8k/edit?usp=drive\\_web](https://docs.google.com/presentation/d/1-PgaZBb0Ol4FeK80SZQzfrE5FxyHMWhLLaiyXO6g8k/edit?usp=drive_web)[6]

[https://docs.google.com/presentation/d/1diA6U\\_LP1nGZLIJdz4Ja5gQ09JuTiQ024u8vhZJWEwc/edit#slide=id.g1df030e786\\_0\\_0](https://docs.google.com/presentation/d/1diA6U_LP1nGZLIJdz4Ja5gQ09JuTiQ024u8vhZJWEwc/edit#slide=id.g1df030e786_0_0)[7]

[https://docs.google.com/presentation/d/1CBmKnMQ0Fp6Hn3GgunB68y1evh0RNRBZFp3oXihHqzl/edit#slide=id.g1f364b7511\\_2\\_5](https://docs.google.com/presentation/d/1CBmKnMQ0Fp6Hn3GgunB68y1evh0RNRBZFp3oXihHqzl/edit#slide=id.g1f364b7511_2_5)[8]