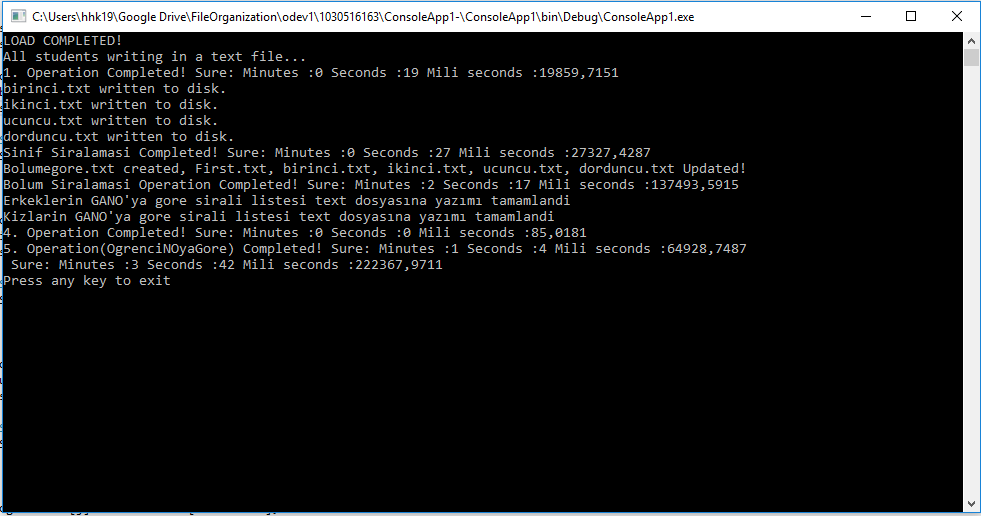
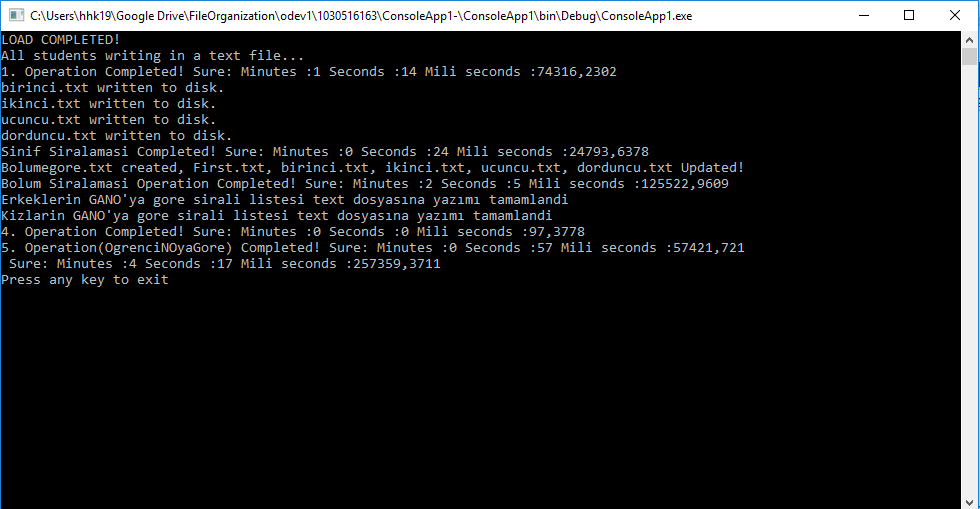
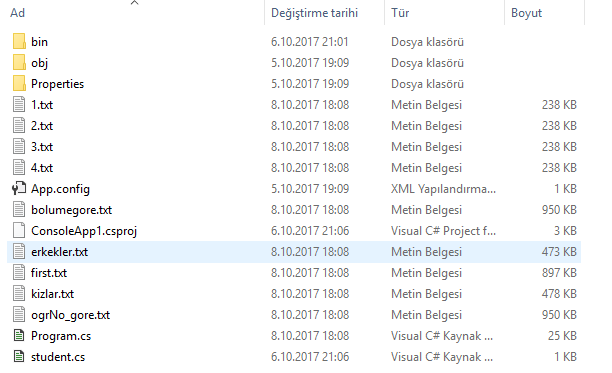
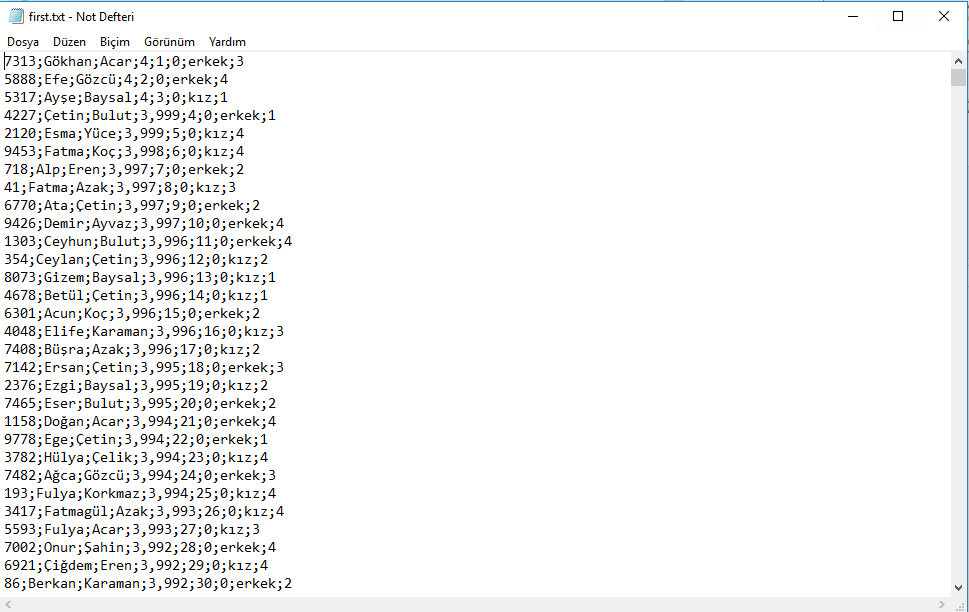
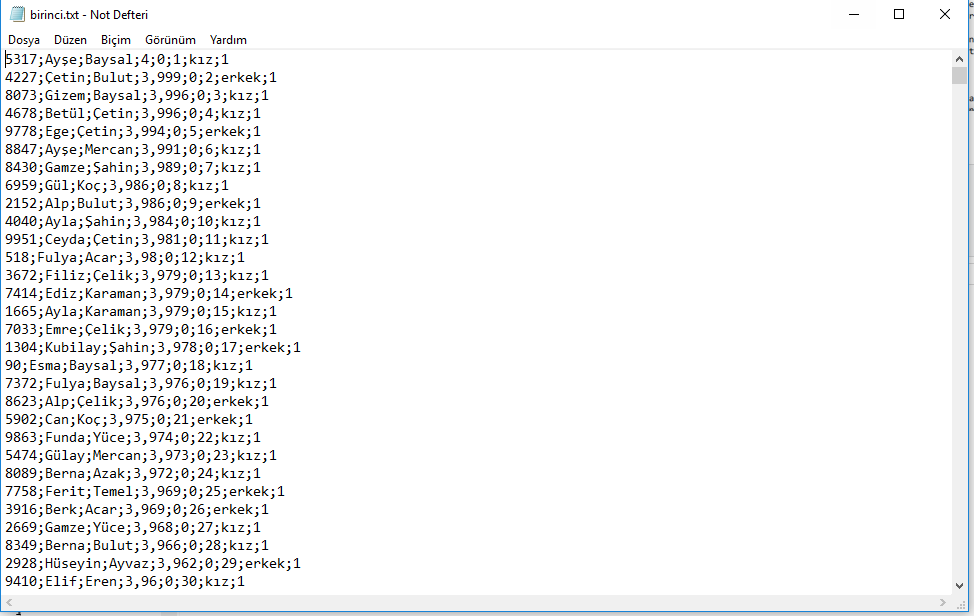
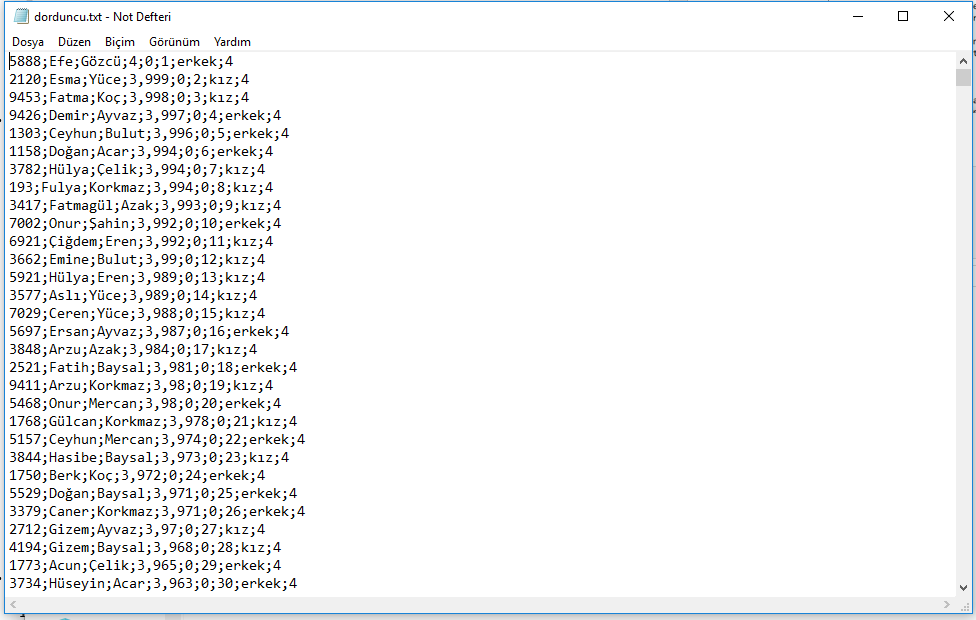
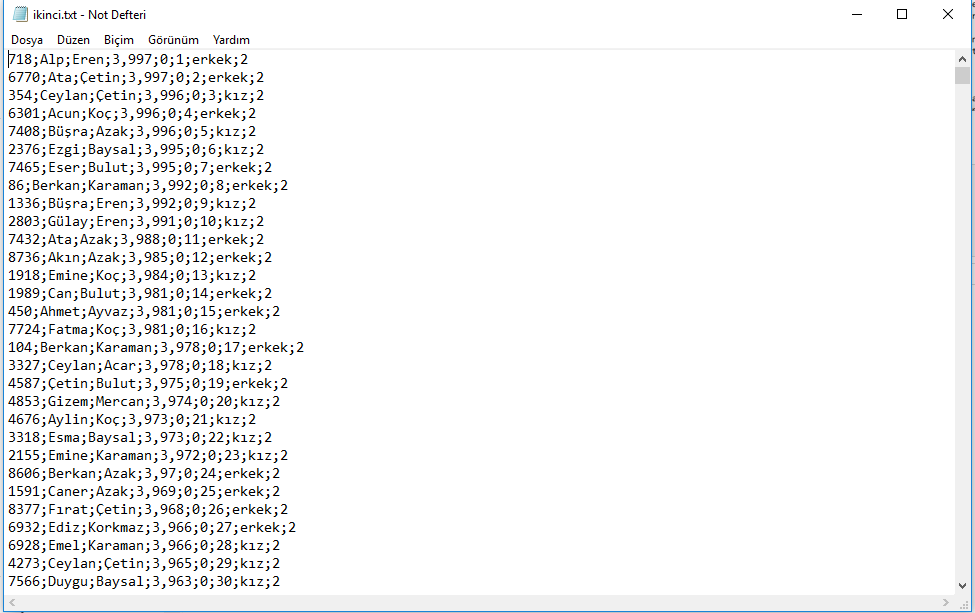
Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümü Dosya Organizasyonu Teknikleri Dersi 1. Ödevi(Düzeltildi)

* Bu ödevi yaparken programlama dili olarak C# tercih ettim.
* Bu raporda ekran çıktılarını ve text dosyalarından alınan ekran görüntülerine erişebilirsiniz.
* Bu ödevde öğrenciler için key value olarak değer tutmak istedim. Bundan dolayı öğrenciler (student) diye bir class tanımlayıp içerisine istenen alan bileşenleri tanımladım.
* Bu class yapısının içerisinde String olarak isim, soyad, cinsiyet bilgilerini tanımladım.
* Öğrenci numarası için int olarak ogr\_no tanımladım.
* Gano içinse float isteniyordu ilk başta float olarak GANO tanımladım. İlerleyen bölümlerde sıkıntı çıkardığı için double olarak tanımladım. Virgülden sonra 3 hane gösterilmesi isteniyordu onu da ayarladım. Sonuçta GANO istendiği gibi elde ediliyor.
* Bölüm sırasını, sınıf sırası ve sınıf alanlarını da int olarak tanımladım. Bu alanlar student sınıfının içerisinde bolum\_sira, sinif\_sira ve sınıf olarak tanımlı.
* Son olarak student class yapısında her bir alan için getter ve setter tanımladım ki bunları fazlasıyla kullandım.
* Esas bütün operasyonların döndüğü sınıf “program.cs” adlı class yapısında tamamlanıyor.
* Programda her bir operasyonun ne kadar sürede gerçekleştirildiğini hesaplamak için Stopwatch() sınıfından tanımlı bir nesne kullandım. “sw” her bir adımın ne kadar sürede gerçekleştirildiğini hesaplamak için , “sw2” ise bütün programın baştan sona ne kadar sürede tamamlandığını hesaplamak için kullandım.
* Oluşan listeleri de dosyaya yazmak için StringBuilder sınıfından yararlandım.
* Her bir dosyada ilk olarak Ogrenci No , isim , Soyad, GANO, Bölüm sıra, Sınıf Sıra, Cinsiyet, Sınıf sütunları bulunuyor.
* Delimiter olarak yani sütunları birbirinden ayırmak için “;” kullandım.
* Bütün öğrenciler “ogrenciler” adlı bir listede bulunmakta. Bu listenin türü ise “student”.
* Bütün öğrenciler “first.txt” adlı text dosyasında bulunuyor. Kullandığım fonksiyonlarda da bu text dosyasını sıkça görebilirsiniz.
* “birinci.txt” adlı text dosyasında birinci sınıf öğrencileri GANO sırasına göre bulunmakta.
* “ikinci.txt” adlı text dosyasında ikinci sınıf öğrencileri GANO sırasına göre bulunmakta.
* “ucuncu.txt” adlı text dosyasında ucuncu sınıf öğrencileri GANO sırasına göre bulunmakta.
* “dorduncu.txt” adlı text dosyasında dorduncu sınıf öğrencileri GANO sırasına göre bulunmakta.
* Erkek isimlerini “erkekler” adlı bir String listeden alıyorum.
* Kızların isimlerini “kizlar” adlı yine bir String listeden alıyorum.
* Bu isim listelerinin içini kendim internette bulduğum isimlerle doldurdum.
* “cinsiyet” adlı bir String türünde listede cinsiyetleri tuttum.
* “soyad” adlı String listede ise çeşitli soy isimleri yer alıyor.
* “sinif” adlı int listede 1,2,3,4 sınıflar yazıyor.
* “ogrNo” adlı int listede ise bütün öğrencilerin numarası yer alıyor. Bu listeyi öğrencilere numara atamak için kullandım.
* İlk olarak “ogrNo” adlı listeye 1 den başlayıp 10000’e kadar numaraları ekledim.
* İkinci for döngüsünde ise öğrencilerin alanları dolduruluyor.
* Bu for döngüsünün içerisinde ilk olarak student class yapısından bir ogr adlı bir nesne tanımladım.
* Erkeklerin bulunduğu isim listesinden bir random index alıyorum. Bu yöntemi cinsiyet, kızlar, soyad ve ogrNo içinde kullandım. Eğer cinsiyet erkek ise öğrencinin ismini erkekler[index] yazarak erkekler listesinden, kız ise kizlar[index] ile kızların bulunduğu listeden aldım.
* Soyad için bir kontrol yapmaya gerek yok. Random olarak bir index seçilip soyad[index] ile öğrencinin soyad ataması yapılıyor.
* Öğrencini numarası da rastgele seçildi ve ogrNo[ogrNoindex] ile atandı. Buradan sonra derste de gösterildiği gibi her bir öğrencide bir numara olması için yani unique yapıda olması için ogrNo listesinden RemoveAt(ogrNoindex) ile o listeden atanan numarayı sildim. Bu sayede her öğrencinin kendine özgü bir tane numarası bulunacak.
* Gano için 1 ile 4 arasında random bir numara üreten fonksiyon kullandım. Burada karşılaştığım sorun çok tekrarın bulunmasıydı bunun için private static readonly Random getrandom = new Random(); kullandım.
* GetRandomNumber fonksiyonu ile Gano için random bir sayı ürettim. Virgülden sonra 3 haneli olacak şekilde GANO üretildi.
* Yukarıda random index seçtiğimi söylemiştim bunun içinde GetRandomNumber2 adlı fonksiyonu kullandım. Bu fonksiyonda min ve max değerleri girilip random bir int dönderiliyor.
* Bundan sonra alanbilgileri doldurulan öğrenci “ogrenciler” adlı bir listeye ekleniyor.
* Şimdi de öğrencilerin sınıf atamasını yapmamız gerekiyor. Bunu için 0 dan başlayıp 10000’e kadar devam eden bir for döngüsü tanımladım. 10000 öğrenci olduğundan dolayı.
* Sınıf listesinde random bir index geliyor ve if else yapısıyla bütün sınıflar kontrol ediliyor. Mesela gelen sınıf 1 ve mevcut<2500 ise öğrenciler[j].Sınıf=sinif[sinifindex] ile atanıyor. birinci.Add(öğrenciler[j]) ile de o öğrenci o sınıfın listesinde eklenmiş oluyor.
* Diyelim mevcut doldu bu durumda else in içerisinde girecek ve bütün sınıflarda 2500 öğrenci olmayacak. Ben bu durumla çok uğraştım. Bunun içinde else yapısının içerisinde ”j—“ yaptım yani o döngü hiç olmamış kabul etsin ki her sınıfta 2500 öğrenci toplamda 10000 öğrenci bulunsun.
* Bolum sıralaması için Bolumsirala2 adlı bir fonksiyon kullandım. Bu fonksiyona String olarak bütün öğrencilerin bulunduğu text dosyasının ismini değişken olarak tanımladım.(String ogrenciler). Liste yapısından faydalanmak istediğim için “sinif” adında boş bir liste tanımladım.
* System.IO.File.ReadAllLines(@dosyayolu) ile bütün öğrencilerin bulunduğu text dosyasını satır satır okumak için string[] lines tanımladım.
* Daha sonra bu okuduğum dosyanın güncel halini yazmak için StringBuilder yapısı tanımladım.
* Dosyayı güncellemek için listede olan bir item’ı satır satır appendline diyerek ekledim.
* En sonda da StringBuilder yapısını String’e dönüştürüp System.IO.File.WriteAllText(@dosyayolu) ile yazdım.
* 2 For döngüsü ile başladım. İlkinde satır satır okuyup “;” sayesinde line.Split(‘;’); ile field’lara böldüm. Student class ından nesne tanımlayıp bütün satırları bu şekilde class yapısına oturttum. Başta student türünde boş bir liste oluşturmuştum. Bütün bir satır bir öğrenci olacak şekilde bu listeye ekleme yaptım.
* İkinci for döngüsünde ise Sequential file organization yapısına uygun olmasına dikkat ettim. Satır satır okuma ve yazma yapıyoruz. Sinif.OrderBydescending yaptım. Sadece sinif[0] ile en üstte yer alan satırı almış oldum. Bu kaydı StringBuilder’a append yaptım. Sonra bu kaydı sinif listesinden removeat(0) ile sildim. Bu şekilde en üstteki değer en yüksek değer olmuş olacak ve satır satır işlem yapabileceğiz. En sonda da System.IO.File.WriteAllText(@dosyayolu) ile ilk başta okuduğumuz bütün öğrencilerin olduğu text dosyasını güncellemiş oldum.
* Bu aşamadan sonra 1. Operasyonu yapabiliriz. Yine bir for döngüsüyle bütün öğrencilerin bilgilerini StringBuilder nesnesine ekliyoruz. System.IO.File.WriteAllText ile de text dosyası olarak yazdırdım.
* Console.WriteLine() ile konsola bu aşamanın tamamlandığını ve ne kadar sürede tamamlandığını gösteren satırı yazdırdım. Süre için StopWatch class kullandım.
* 1.Aşama da bütün öğrenciler oluşuturuldu.
* 2. Aşama da bu öğrencilerin sınıf listesi oluşturuluyor.
* İlk Başta bütün öğrencilerin bulunduğu text dosyası kullanılarak sınıf listesi oluşturuluyor. Bu listelerde öğrenciler GANO’ya göre sıralı bir şekilde bulabilirsiniz.
* Bunun içinde Sinifsirala2() adında (string v) şeklinde parametre alan bir fonksiyon oluşturdum.
* Bu string parametre ile bütün öğrencilerin bulunduğu text dosyasının ismini aldım.
* Boş sinif adında boş bir liste oluşturdum.
* System.IO.File.ReadAllLines(@dosyayolu) ile string[] lines’a satırları atadım.
* StringBuilder tanımladım. Yine bölüm sıralamada kullandığım mantığı kullandım. 2 tane for döngüsü bunun için yeterli oldu. İlk for döngüsünde student tipinde nesneye okuduğum alanları atadım. ‘;’ delimiter olarak kullanıldı. Bu nesneleri de ilk başta oluşturduğum boş listeye ekledim.
* İkinci for döngüsünde ise tekrar okuduğumuz satır sayısı kadar olacak şekilde; sinif listesini orderbydescending(x=>x.Gano) ile güncelledim. Ama ilk başta yer alan kaydı sinif[0] ile sinif sirasını “i” döngü sayacını kullanarak atadım. StringBuilder a bu nesneyi appendledim. Sinif.RemoveAt(0) ile ilk başta yer alan kaydı sildim. Döngünün bir sonraki adımında bu işlemleri tekrarlayarak teker teker bütün satırlarda sıralayıp text dosyasını güncelliyor. System.IO.File.WriteAllText(@dosyayolu) ile String’e dönüştürdüğüm StringBuilder’ı yazmış oldum.
* 2. Aşamanın sonunda Sınıf listelerini GAnoya göre sıralı bir şekilde elde ettik.
* 3. Aşamanın sonunda ise Bölüm sırasına göre bütün öğrencilerin bulunduğu listeyi elde ettik.
* 4. Aşama da cinsiyete göre 2 farklı liste isteniyordu. Bunun için ilk olarak Cinsiyetegore2 adlı bir fonksiyon tanımladım. Bu fonksiyona bütün öğrencileri dosyadan okuyup eklemek için hepsi adında boş bir liste oluşturdum. System.IO.File.ReadAllLines(@dosyayolu) ile text dosyasını okudum. 2 farlı StringBuilder tanımladım. Birisi erkekler için ikincisi de kızlar için. İlk for döngüsünün içerisinde hepsi isimli bütün öğrencilerin bulunduğu listeyi satır satır okuyarak doldurdum. İkinci for döngüsünde ise satır sayısı kadar olacak şekilde if, else yapısı kurdum. Burada listede yer alan öğrencinin cinsiyeti “erkek” ise birinci stringBuilder ‘a appenledim. Diğer durumda ikinci stringbuilder’a appenledim. En sonda da System.IO.File.WriteAllText ile erkekler.txt, kızlar.txt adlı 2 text dosyası oluşturdum.
* Bu aşamadan sonra ekrana yine 4.aşamanın tamamlandığını ve ne kadar süre de tamamlandığını gösteren bir mesajı konsola yazdırdım.
* Son aşamada içinde OgrenciNoyaGore2 adlı bir fonksiyon oluşturdum. Bu fonksiyona da yine bütün öğrencilerin ye aldığı “ogrenciler” adlı boş bir listeyi tanımladım. Bu fonksiyonda diğerlerinde olduğu gibi (String v) parametresi alarak işleyen bir fonksiyon. Bütün öğrencilerin bulunduğu dosyanın ismini alıyor. Boş bir listeyi satır satır for döngüsüyle okuyarak student yapısında olan boş listeyi doldurduk. İkinci for döngüsünde ise Ogrenci numarasına göre sınıf sıralama ve bolum sıralama esnasında da anlattığım algoritmayı kullanarak sıraladım. Son olarakta System.IO.File.WriteAllText ile de ogrNo\_gore.txt olarak kaydettim.
* Console.WriteLine() ile konsola bu aşamanında tamamlandığını ve ne kadar sürede tamamlandığını gösteren mesajı yazdırdım.
* Yaptığım iyileştirmelerden sonra Bütün işlemler 3 dk 40 sn ile 4dk 40 sn. arasında tamamlandı.
* Şimdi program çıktılarına bakalım;
* Programın konsol çıktısı;
* 
* Yukarıda yer alan programın konsol ekranında her bir aşamanın ne kadar sürede tamamlandığını görebilirsiniz.
* Aşağıda yer alan resimde projenin dosyalarını görebilirsiniz. Burada her bir aşamada istenen listeler text dosyası olarak kaydedildi.
* 
* 

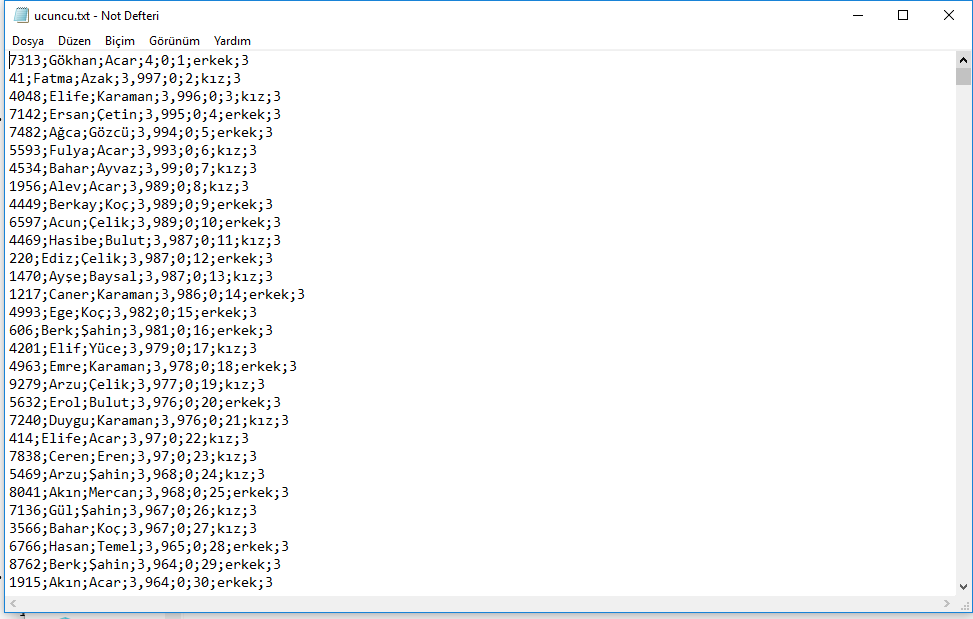
1.Aşamada istenen liste ve görüldüğü üzere daha hiç kimsenin bölüm ve sınıf sırası bulunmuyor. Rastgele öğrenci numaraları, isim, soy isim, sınıf ve gano. Tamamlanma süresi

Yaklaşık olarak 21 sn.

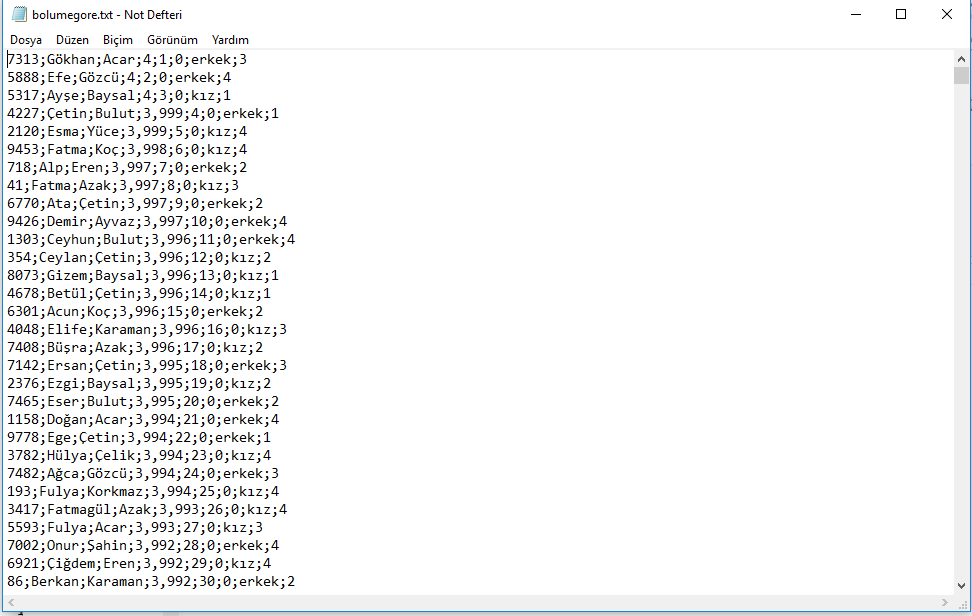


Birinci sınıfların GANO ya göre listesiDördüncü sınıfların GANO ya göre listesi

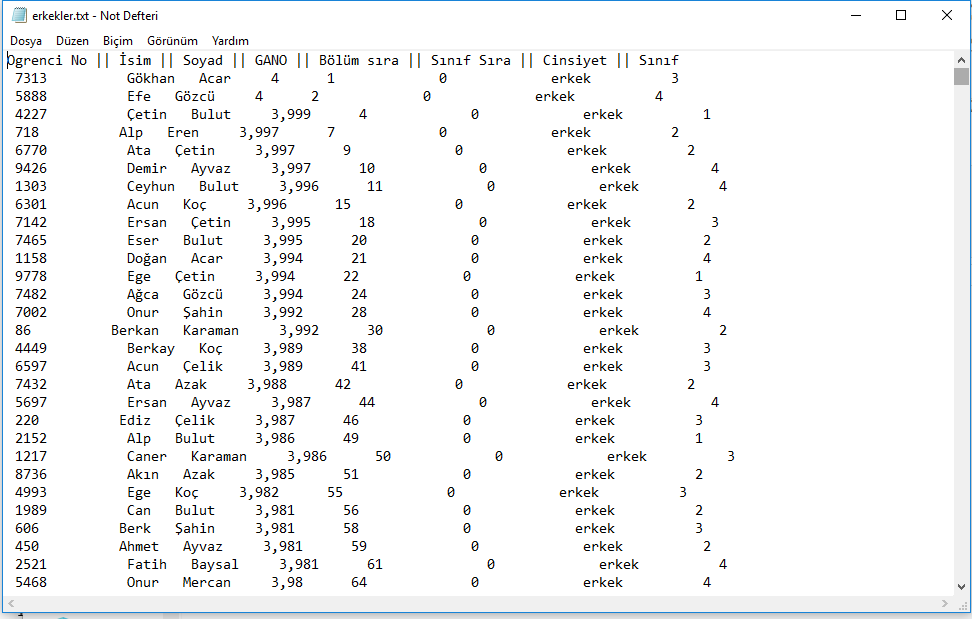
İkinci sınıfların GANO ya göre listesi



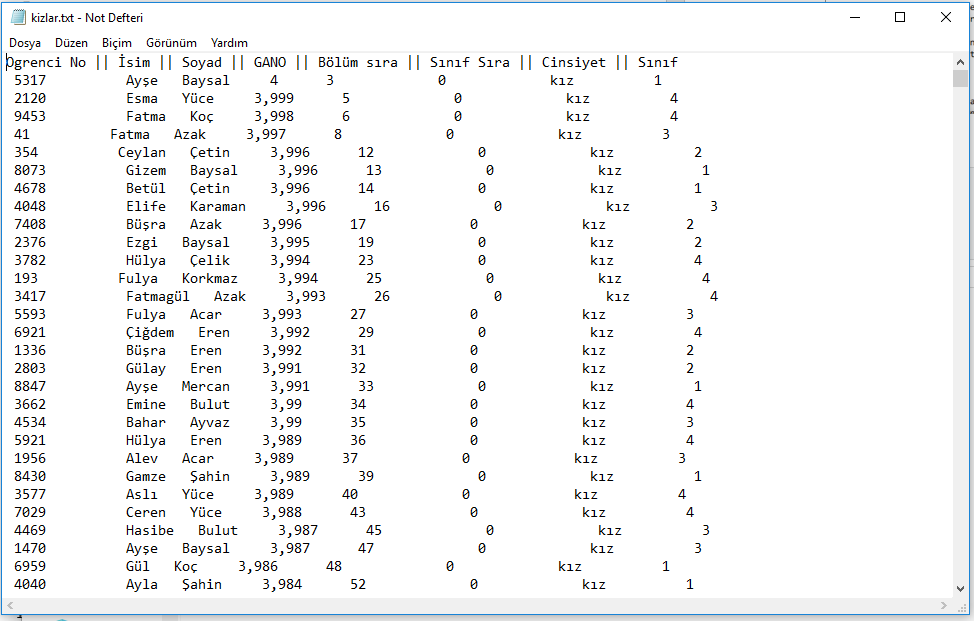
Üçüncü sınıfların GANO ya göre listesi



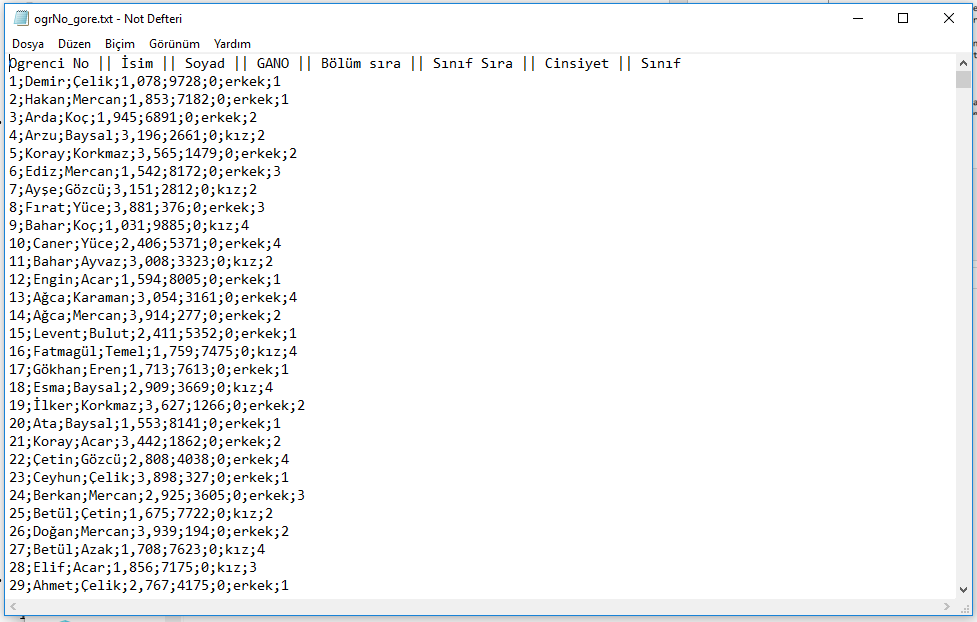
Bölüm sırasına göre öğrencilerin listesi Gano ile paralel olarak listenin ilerlediğini görebilirsiniz. 3. Aşamanın tamamlanma süresi 2 dk. 17 sn.



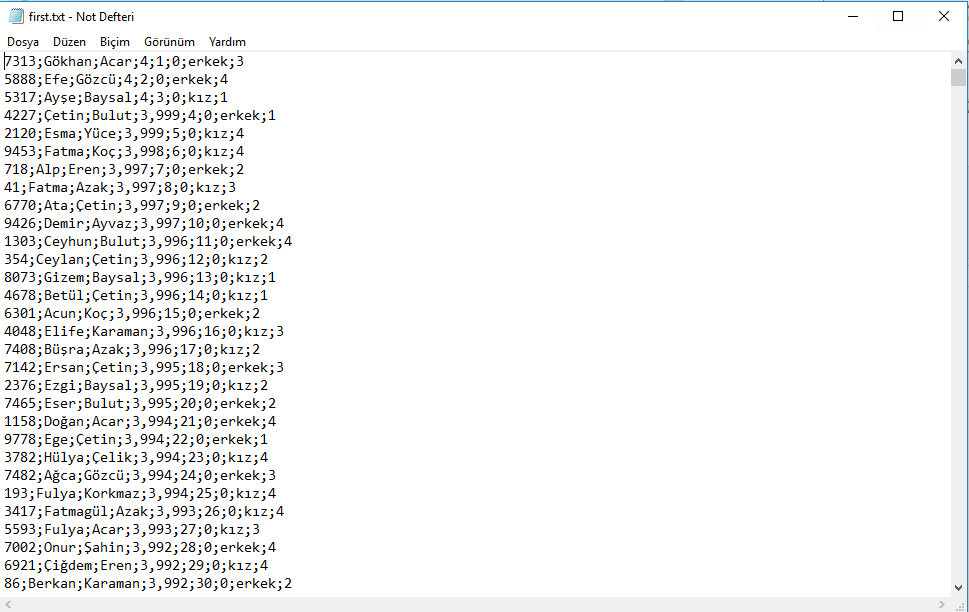
Cinsiyete göre listeler; ilk olarak erkeklerin listesi. Cinsiyeti erkek olup Ganoya göre sıralama yaptım.



Cinsiyeti kız olan öğrencilerin Ganoya göre listesi. 4. Aşamanın tamamlanma süresi: 85 ms. Çok hızlı oluyor. Sadece ayırma yaptığımdan dolayı.



Öğrencilerin, öğrenci numarasına göre listesi. 5. Aşamanın tamamlanma süresi: 57 sn. Neredeyse 1 dk.



Toplam geçen süre: 3 dk. 42sn – 4 dk. 17 sn.