



***BLM 103- Algoritmalar ve Programlama I
2019-2020 Güz Dönemi***

RLE Veri Sıkıştırma

***Proje Teslim Raporu
13 Ocak 2020***

Berk Tunç, Cüneyt Balcı

1	GİRİŞ	1
1.1	Projenin Amacı.....	1
1.2	Proje Ekibi.....	1
2	GELİŞTİRİLEN UYGULAMA	2
2.1	Kullanılan Araçlar	2
2.2	Tasarım	2
3	SONUÇLAR.....	3

1 Giriş

1.1 Projenin Amacı

Dosya boyutunu küçültmek amacıyla birbirini tekrarlayan değerlerin, bu verilerin bir örneği ve değerin kaç kez tekrarlandığı yan yana yazılarak sıkıştırma yapılmasıdır.

1.2 Proje Ekibi

Adı Soyadı: Cüneyt Balcı

Numarası: 190301019

Özgeçmişi: 28.08.2000 yılında İstanbul'da doğdu.2018 yılında Maltepe Final Temel Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. C ve Java ve Python dilleriyle ilgileniyor.

Adı Soyadı: Berk Tunç

Numarası: 190301003

Özgeçmişi: 04.02.2000 yılında doğdu. 2018 yılında Şehit Osman Altinkuyu Anadolu Lisesinden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde Lisans eğitimi almakta. C,C#,HTML dilleriyle ilgileniyor.

2 Geliştirilen Uygulama

2.1 Kullanılan Araçlar

-Visual Studio 2019
-Powerpoint
-Word

2.2 Tasarım

Bu kodda kütüphaneleri girerek mainden kodu başlatıyoruz. Yüksek sayılı bir char dizisi tanımlıyoruz. Böyle yapmamızın sebebi eğer dosya yüksek verili bir dosya olursa overflow olmaması içindir. Sonrasında FILE komutuyla dosya oluşturuyoruz. count ve sikistircoz isimlerinde iki integer değişken tanımlıyoruz. Burada count değişkeni bastırılacak harfin kaç tane olduğunu saymak için atanmıştır. Sonrasında kullanıcıdan sıkıştırma veya çözme yapması için komut vermesi için kullanıcıdan giriş alıyoruz.

Eğer 1 girilirse dosya sıkıştırılmaya başlanacak ve dosya okunmak için açılacak. Sonrasında 0'a eşit bir i değişkeni integer olarak ve char1 değişkeni char olarak tanımlanacak. 15.satırda while döngüsü içinde okunan değerler char1 değişkenine ve sonrasında arr dizisine atanmak üzere dosyadan EOF(End Of File) komutunu görene kadar arr dizisi içinde i değişkeni ile beraber i artırılarak okunuyor. Burada EOF komutu dosyanın sonu anlamına gelir. Sonrasında da dosya kapatılır.

21. satırda dosya tekrardan yazma işlemi için açılıyor. Bir for döngüsü açılarak k değişkeni 0 olarak tanımlanıyor ve dosyadan okunan harfler i'ye kadar gittiği için k, i'ye kadar artırılarak devam ediyor count'a 1 atanıyor. Daha sonrasında while döngüsüne girilerek k+1'in i-1'den küçük olduğu ve dizinin bi sonraki elemanının bi önceki elemanına eşit olduğu zamanlarda k ve count 1 artırılır. While döngüsü çıkışında eğer diziye devam ederken boşluk görürse sıradaki karaktere gelip sıkıştırmaya başlamaktadır. 32. Satırda da dosya kapatılmaktadır.

34. satırda da eğer 2 girişi alınırsa çözme işlemine başlanacaktır. 35'ten 38'e kadar olan satırlarda da sıkışmış dosya okunurken sayıyı alacak arr1 dizisi, harfi alacak arr2 dizisi, harfin atanacağı char2 integer'ı, sayının tanımlanacağı num2 integer'ı, scan edilen değer alınacağı a integer'ı tanımlanmaktadır. Sonrasında while döngüsü içine girilip sıkıştırılan dosyadan kaç kez sıkıştırıldığı ve o harf alınıp num2 ve char2 ye dosya bitimine kadar(EOF ile) atanmaktadır. Sonrasında alınan harfler sayı ve harf için oluşturulan dizilere kaydedilmektedir ve a 1 artırılmaktadır. Ardından dosya kapatılmaktadır.

46. satırda fopens komutu ile dosya yazma işlemi için açılır. For döngüsü açılarak içinde j int olarak tanımlanıp 0'a eşitlenmektedir ve a'ya eşit olana kadar artırılmaktadır. Ardından while arr1[j] 0'a eşit olmadığı durumlarda döngüye girilir. Sonrasında fprintf komutu ile karakter dosyaya yazılarak çözülmeye başlanır. 50. satırda ise okunan sayı 1 azaltılarak kaç harf yazılacağı hedeflenmektedir. 53. satırda da dosya kapatılır. Eğer 1 veya 2 den farklı değer girildiyse program durur ve kapanır.

Bu kodda herhangi bir eksik bulunmamakta fakat karşılaştığımız bir sorun; sıkıştırma yaparken \n görmesi sebebiyle satır sonuna 1 bastırmasıdır.

3 Sonular

Bu proje iin ncelikle neler yapabileceğimizi stnkr dřndk ve kısım kısım kodumuzu oluřturmaya bařladık. ncelikle sıkıřtırmayı hedefledik. Bu hedefimizi birkaç problem yařadıktan sonra (stack overflow, sıkıřtırmayı kod kaynaklı hatalı yapması) daha fazla emek vererek gerekleřtirdik. Daha sonrasında ise kodumuza zmeyi ekledik. Bu kodun mantıėını anladıėımız iin sıkıřtırmaya oranla daha kolay bir ařamaydı bizim iin. Her eklemekten sonra srekli test ettik ve kodumuzu yapabildiėimiz kadar stabilleřtirmeye alıřtık. Kodu direk olarak yazmaktansa kısım kısım yazdıėımız iin iřimiz ok daha kolaylařtı. Kod yazmada ekip alıřmasını kk aplı olsa da yařamıř, deneyimlemiř olduk.

Hazırlanan sunum video'su adresi: <https://www.youtube.com/watch?v=ptir-Scy2WY&t=52s>

Dosyaların github adresi: <https://github.com/cuneytbalci/FBU-RleVeriSikistirma>