***3 Boyutlu Görüntü Dosyaları Projesi***

***3D Image Files Project***

Cumali TOPRAK

Mühendislik Fakültesi, Bilgisayar Mühendisliği

Kocaeli Üniversitesi

Kocaeli, Türkiye

cumalitoprakk@gmail.com

Berkay Efe ÖZCAN

Mühendislik Fakültesi, Bilgisayar Mühendisliği

Kocaeli Üniversitesi

Kocaeli, Türkiye

berkayefeozcan@gmail.com

*Özetçe*—Günümüzde bilişim ortamlarında herhangi bir veri depolama ortamlarından veri okumak bu ortamlara verileri kaydetmek çok önemli bir hal almış durumdadır. Bu ortamlarda bir çok veri tabanı bulunmaktadır. Biz de bu raporun konusu da olan 3 boyutlu dosyalardan belirtilen koşullara göre verileri çekip, bu verilerle sırası ile bir dizi işlem gerçekleştiren bir yazılım oluşturmuş bulunmaktayız. Bu yazılımın amacı sayısı belirli olmayacak şekilde dosyadan, şayet dosyada herhangi bir problem yok ise, başlık bilgileri dediğimiz ilgili dosyanın ilk beş satırında belirtilen koşullara göre dosyanın içerisinde bulunan 3 boyutlu noktaları varsa renk bilgileri ile birlikte dosyadan çekmektir. Sonrasında kullanıcının bu noktalara göre talep ettiği işlemleri ona sunmaktır. Yazılımımızı sınadığımız zaman görüyoruz ki bu işlemleri gayet başarılı bir şekilde gerçekleştirmektedir.

Anahtar Kelimeler—Görüntü Dosyaları, Ascii Formatlı Dosya, Binary Formatlı Dosya, Dosya Okuma, Dosyaya yazma.

*Abstract* — Nowadays, it is very important to read data from any data storage media in IT environments and to save data to these media. There are many databases in these environments. We have created a software that extracts the data from the 3D files that are the subject of this report according to the specified conditions and performs a series of transactions with these data respectively. The purpose of this software is not certain number of files, if there is no problem in the file, according to the conditions mentioned in the first five lines of the file we call the title information, if there are three points in the file with color information to draw from the file. After that, the user requests the transactions according to these points. When we test our software, we see that it performs these operations very successfully.

Keywords— Image Files, Ascii Format File, Binary Format File, Reading From File, Writing to File.

1. GİRİŞ

Veri havuzunun bu denli genişlediği bu teknoloji çağında elbette bu verileri saklamak, gerektiğinde bu verileri kullanmak veya bu verilerde değişiklik yapabilmek için birçok veri saklama ortamları kullanılmaktadır. Bu veri depolama ortamlarından biri de dosyalardır. Biz de bu raporda da değinmiş olduğumuz projede iki farklı dosya türünde bir takım işlemler yaptık. Projemizin detaylarına inersek, bu projede programımızın bulunduğu klasörle aynı klasöre konan, sayısı ve dosya türü belli olmayan dosyalardan verileri çekip belirtilen koşullara göre işlemler yapmaktır. Programı çalıştırdığımız zaman öncelikle bu programın sadece .nkt uzantılı dosyaların sayılarını belirleyip bu dosyalar üzerinde belirtilen işlemleri yapması beklenmektedir. Şayet programımız istenilen uzantıdaki dosya türünden farklı dosyaları da bu işleme tabii tutarsa bu karşımıza büyük bir yazılım açığı olarak çıkacaktır. Bu konuda yapmış olduğumuz kontroller neticesinde bu problem ortadan kalkmıştır. İstenilen formattaki dosyalar belirlendikten sonra tüm dosyaların kontrolü yapılacak, eğer herhangi bir dosyada bir bozukluk, format hatası, eksik veri varsa bu dosyalar işlem dışı kalacaktır. Dosyaların kontrolü de yapıldıktan sonra raporun üçüncü kısmında detaylı bir şekilde açıkladığımız bir takım işlem uygulanacak bu işlemlerin sonuçları her bir dosya için çıktı dosyasına yazdırılacak şekilde tasarlanmıştır. Programımızdan en iyi verimi almak için en iyi algoritmalar kullanılmaya çalışılmış, bu algoritmalar da raporun ilerleyen bölümlerinde detaylı bir şekilde açıklanmıştır.

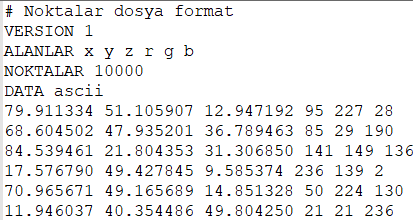
1. TEMEL BİLGİLER

## Veri Nedir?

Günlük işlerimizin tamamını yaparken adeta bilgi yağmuru ile karşı karşıya kalırız. Beynimiz ise bu bilgi yağmurundan işine yarayacağını düşündüklerini alır yorumlar ve biz de kararlarımızı bu yorumlara göre veririz. Hatta zihnimiz daha sonra kullanmak için bazı bilgileri saklar. Sizce, evden okula yürüyerek gelen bir öğrenci ne tür bilgilerle karşılaşmaktadır? Örneğin, Saatimize bakarak geç kalıp kalmadığımızı kontrol ederiz. Eğer gecikmişsek daha hızlı yürürüz. Veya, Evden çıktığımızda hava ile ilgili bir fikrimiz olur. Hava beklediğimizden daha soğuksa eve tekrar dönüp daha kalın bir mont alabiliriz. TDK’ye göre ise veri; 1-Bir araştırmanın, bir tartışmanın, bir muhakemenin temeli olan ana öge, muta, done. 2- Olgu, kavram veya komutların, iletişim, yorum ve işlem için elverişli biçimli gösterimi olarak tanımlanır. [1]Bununla birlikte bizim asıl çalışmalar yaptığımız bilgisayar bilimlerinde de ayrıca verinin bir tanımı vardır. Bilgisayarlar da tıpkı beynimiz gibi çeşitli kararlar vermek veya işlem yapabilmek için bilgi girişine ihtiyaç duyar**. Bilgisayarların sonuca ulaşabilmek için algıladığı, işlediği, sonuç ürettiği veya daha sonra kullanmak üzere depoladığı her şeye veri denir. İşte bu veriler de çoğu zaman dosyalarda farklı formatlarda saklanırlar. Bu olaya da veri formatlama denir. Biz yapmış olduğumuz bu projemizde ascii ve binary olmak üzere iki farklı dosyadan okumalar yapıp bu dosyalar üzerinde bir takım işlemler yaptıktan sonra istenilen formatta çıktılarımızı formatladık.**

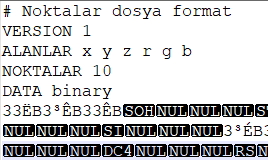
## Ascii Formatlama

Bir ASCII dosyası herhangi bir platform ya da işletim sistemi tarafından kolayca tanınan ve okunacak bir belge sağlayan bir metin dosyasıdır. Bu temelde herhangi bir kısıtlama olmaksızın dosya daha erişilebilir hale getirir ve sık ASC dosya uzantısını kullanır. ASCII Bilgi Değişimi için Amerikan Standart Kodu ve standart bir karakter seti ve 1960 tapedrives ve tele yazıcılar gibi diğer aygıtlar gibi bilgisayarlar için tasarlanmıştır[2]. Bugün, ASCII kodları şu anda iletişim ekipmanları, bilgisayar ve metinler istihdam diğer cihazlar her türlü kullanılmaktadır. Onun özellikleri 128 karakter seti 7-bitlik karakter içerir. Ayrıca A 0'dan Z ve sayılar 9 ila üst ve alt harf içerir. Bugün modern bilgisayarların internet üzerinden bilgi gönderme karakter ASCII kümesi kullanmak ve HTML web sitesi oluşturma özelliğidir. Tarayıcınız tarafından erişilebilir internet sayfaları ASCII kodları yazılmış HTML olmadan düzgün çalışamaz. Biz de aşağıda resmini verdiğimiz gibi ascii formatlı dosyasındaki 3 boyutlu noktaları kullanarak işlemler gerçekleştirdik.



Şekil-1: Ascii Formatında verilmiş dosyadaki 3 boyutlu noktaların gösterimi.

## Binary Formatlama

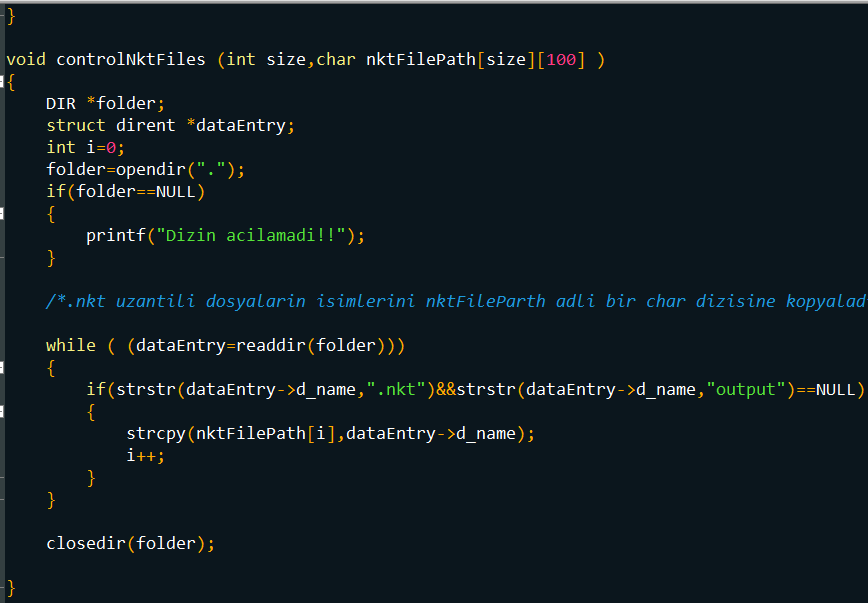
İkili format, dosya bilgilerinin olanlar ve sıfırlar şeklinde veya başka bir ikili (iki durumlu) sekansta saklandığı bir formattır. Bu tür bir format genellikle bilgisayar programlaması ve bellekteki çalıştırılabilir dosyalar ve sayısal bilgiler için kullanılır. Matematiksel anlamda, ikili, çok basamaklı sayıları veya diğer bilgileri bir ve sıfır dizisine dönüştürür. Bazıları bunu açma-kapama formatı olarak adlandırabilir, çünkü her veri biti iki durumdan birindedir.[3] Toplu olarak, bunlardan ve sıfırlardan oluşan dizeler (veya açık ve kapalı gösterimler) çok daha karmaşık veri kümeleri oluşturabilir. İkili formatta olan dosyalar genellikle makine dilinde sunulduğu şekilde tanımlanabilir. Bilgisayarlar bu ikili olanları ve sıfırları alır ve bunları çalıştırılabilir dile veya diğer kod türlerine çevirir. Diğer format türleri arasında, tek tek karakterlerin kendi dijital kodları olarak temsil edildiği metin formatını veya tamsayıları veya diğer verileri temsil etmek için bir temel 16 sayı sisteminin kullanıldığı onaltılık formatı içerir. Biz de aşağıda resmini verdiğimiz gibi binary formatlı dosyasındaki 3 boyutlu noktaları kullanarak işlemler gerçekleştirdik.

Şekil-2: Binary Formatında verilmiş dosyadaki 3 boyutlu noktaların gösterimi.

1. YÖNTEM

## İşlem (Okuma, Ekleme) Yapılacak olan dosyaların belirlenmesi

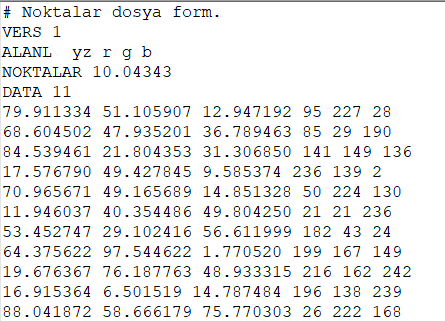
Öncelikle projemizin daha çalışmaya başladığı anda yapması gereken bu işlem sonraki adımların doğru sonucu vermesi için çok önemlidir. Çünkü eğer bu işlem başarısız olursa bundan sonraki tüm işlemler başarısız olacaktır. Bu yüzden dosyalardan veri çekmeye başlamadan evvel kullandığımız dirent.h adlı C dili kütüphanesini projemize ekledik[4]. Bu kütüphane yardımı ile istediğimiz dizine ulaşıp, bu dizinde istediğimiz uzantıda olan dosyaların isimlerini sonradan kullanmak üzere bir diziye atadık. Aşağıda gösterilen algoritma bu dosyaları istediğimiz dizinden nasıl çektiğimizi göstermektedir.



Şekil-3: İstenilen dizinden istenilen dosya uzantına sahip dosya isimlerini çeken algoritma

## Dosyanın Başlık Bilgilerinin Çekilmesi Ve Kontrolünün Yapılması

Projenin bu aşamasında artık gerekli dosyaların gerekli dizinden çekildiği farz edilerek işlem yapılmıştır. Öncelikle her dosya için dosyanın başında beş satırlık bir başlık bilgisi adı altında dosya versiyon bilgisi, nokta türü, nokta sayısı ve veri format türü verilmektedir. Bizden öncelikle bu bilgilerin doğrulamasının yapılmasını, yani herhangi bir dosyanın başlık bilgisinin hatalı olması durumunda bu dosyaların belirlenmesi ve bu dosyalar üzerinde herhangi bir işlem yapılmaması istenmektedir. Bu yüzden, bu dosyaların saptanması için ayrıca bir kontrol mekanizması oluşturulması gerekmektedir. Biz de projemizde bu işlem gerekli adımları sırası ile uygulayarak kontrol algoritmaları gerçekleştirdik. Bu algoritmaya göre eğer herhangi bir dosyanın başlık bilgilerinde hata varsa bu dosya belirlenecek ve işleme sokulmayacak sorunsuz dosyalar ise bir sonraki doğrulama işlemlerinden geçirilecektir.



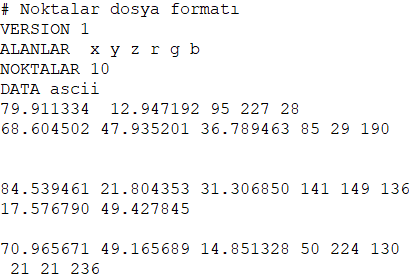
Şekil-4: Başlık bilgileri hatalı verilmiş dosya formatı

## Veri Formatlama Tipine Göre Gerekli Fonksiyon Ve Algoritmaların Uygulanması

Başlık bilgilerinin doğrulamasından geçen dosyalarının türünün belirlenmesi yapılacak işlemlerin doğru sonucu vermesi için elzemdir. Çünkü iki farklı formatlama türü olduğu için bu iki formattaki verileri okumak birbirinden çok farklıdır. Bu yüzden her dosyanın veri formatlama türü belirlenir ve buna göre gerekli fonksiyonlar çağrılıp işlemler yapılır.

## Dosyadaki Nokta Sayısının Ve Nokta Türünün Kontrol Edilmesi

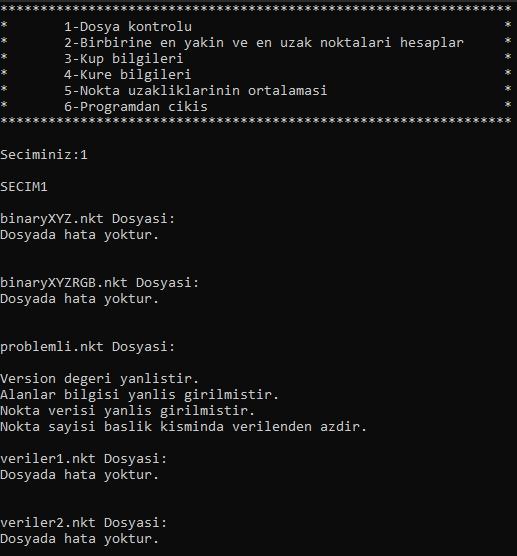
Projenin bu bölümünde dosya başlık bilgilerinde yer alan nokta türü ve sayısı ile bilgilerinin doğrulamasının yapıldığı kısımları içermektedir. Öncelikle Başlık bilgilerinden alınan nokta bilgilerine göre dosyadaki her satır teker teker incelenir eğer noktaların herhangi birinin veya birçoğunun başlık bilgilerinde belirtilen türden farklı türde olması durumunda okuma durdurulup bu dosyanın hatalı olarak işaretlenmesi ve doğrulama sonrası işlemlere sokulmaması gerekmektedir. Bizde bu sebeple projemizde her dosya için ayrı ayrı nokta türü ve sayısı doğrulamasını gerçekleştirdik. Örneğin aşağıdaki şekilde de gösterildiği gibi herhangi bir noktanın x, y veya z konum bilgisinden biri veya daha fazlasının hatalı veya eksik olması durumları göz önünde bulundurulmuş, veyahut eğer nokta türü renk bilgilerini de içeriyorsa bu bilgilerden herhangi birinde bir eksiklik varsa bu dosya hatalı olacağından bunun da kontrolü sağlanmıştır. Aynı şekilde eğer bir veya birkaç satırın boş ise yani herhangi bir nokta bilgisi içermiyorsa bu dosyada hatalı olarak işaretlenmiş ve diğer işlemlere sokulmamıştır.



Şekil-5: Nokta sayısı ve bilgileri yanlış verilmiş dosya formatı

## Hatalı Dosya Var İse Bu Dosyalardaki Hataların Çıktı Dosyasına Yazdırılması

Kullanıcı, yapmış olduğumuz bu programda 1 tuşuna bastığı takdirde tüm dosyaların başlık bilgilerinin ve nokta bilgilerinin doğrulanmasına ait fonksiyonlar çalışmaya başlar. Eğer hiçbir dosyada sıkıntı yok ise çıktı dosyasına “Tüm dosyalar uyumludur” yazdırılır. Ve tüm dosyalar bir sonraki adıma geçerler. Fakat herhangi bir dosya hatalı ise bu dosyaya ait tüm hatalar çıktı dosyasına kendi dosyasının adının bulunduğu konumun altına otomatik olarak yazdırılır. Diğer dosyalar sonraki işlemleri gerçekleştirirken bu dosya diğer işlemlere sokulmaz. Bu sisteme ait şekil aşağıda gösterilmiştir.



Şekil-6: Dosyaların doğrulaması sonucu oluşan çıktılar

## İsterlerin Tanımlanması

Projenin bundan sonraki bölümünde hatalı dosyaların ayıklanması yapılmış, gerekli mesajların çıktı dosyasına yazıldığı farz edilerek işlemlere devam edilir. Hatalı dosyalar işaretlendikten sonra kalan dosyaları kullanarak dört adet işlem uygulanacaktır. Bunlar kullanıcının sırası ile 2, 3, 4 ve 5 tuşlarına basması ile tüm hatasız dosyalara uygulanacaktır.

Ve bunların sonuçları çıktı dosyasına sırası ile yazılacak şekilde program tasarlanmıştır. Bundan sonraki bölümlerde bu adımlar sırası ile detaylandırılacaktır.

## İster 1- Birbirine En Yakın Nokta Ve En Uzak Noktaların Bilgileri Ve Nokta Numaraları

Kullanıcı bu işlemi görmek istediği takdirde her dosya için tüm dosyalardaki noktaların birbiri arasındaki uzaklıklarına bakma ihtiyacı oluşur. Bizde yapmış olduğumuz proje tüm noktaları dosyadan çekip dosya formatlama türüne, nokta sayısına göre bu noktalara bir dizi yardımı ile döngü içerisinde tüm noktaları kendi içerisinde karşılaştırarak birbirine en uzak ve en yakın olan 4 noktayı bulup her dosya için çıktı dosyasına yazdırdık. Bu işlemi yaparken mümkün olduğunca programdan en fazla verim alacak algoritmayı seçmeye özen gösterdik. Programımızın bu isterini sınadığımız zaman gördük ki algoritmamız verilen girdilere göre doğru sonucu veriyor.

## İster 2- Tüm Noktaları İçine Alacak Bir Küpün Kenar Nokta Bilgileri

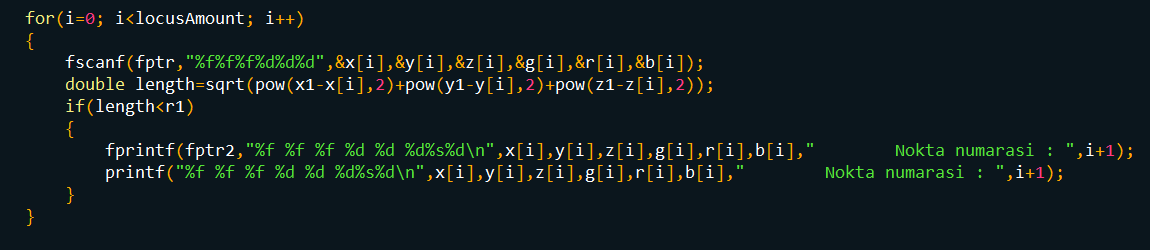
Bu isterin amacı kullanıcı 3 tuşuna bastığında her dosya için sırası ile o dosyadaki tüm noktaları içine alacak bir küpün kenar nokta bilgilerini çıktı dosyasına yazmaktır. Burada dikkat edilmesi gereken şey herhangi bir küp değil en küçük küpün istenmesidir. Bunun biz de projemizde öncelikle bu noktaların içinden maksimun değerde olan x, y, z ve mininum değerde olan x, y, z noktaları karşılaştırarak bulduktan sonra bu noktaların sekiz farklı kombinasyonu ile küpün kenar nokta bilgilerini elde ettik. Aşağıda bu kombinasyon denklemi verilmiştir.



Şekil-7: Tüm noktaları içine alan küpün kenar noktalarını veren kod parçası

*İ. İster 3- Merkezinin 4 Boyutlu Koordinatı Ve Yarıçapı Bilgileri Kullanıcıdan İstenilen Bir Kürenin  
İçinde Kalan Noktaların Bilgileri Ve Nokta Numaraları*

Bu isterde kullanıcı 3 tuşuna bastığında sırası ile kürenin merkezinin x, y ve z koordinatlarını ve yarıçap uzaklığı olan r değerini girmektedir. Girilen bu değerlere göre oluşturulan küpün içinde kalan noktalar tüm dosyalar için hesaplanmaktadır. Her dosya için küpün içinde kalan noktalar çıktı dosyasına sırası ile yazdırılmaktadır. Biz de projemizde bu isteri gerçekleştirmek için öncelikle her dosya için sırası ile tüm noktaların kürenin merkezi ile arasındaki mesafeyi hesaplattık. Sonrasında bu mesafeyi kürenin yarıçapı ile karşılaştırdık. Eğer bu mesafe yarıçaptan küçük ise bu noktanın kürenin içinde kalacağını dolayısı ile şartları sağladığını saptadık ve çıktı dosyasına yazdırdık.



Şekil 8- Kullanıcıdan küre merkez koordinat bilgisi ve yarıçap bilgisi alınan kürenin içinde kalan noktaları hesaplayan kod parçası

## İster 4- Her Bir Noktanın Birbirlerine Olan Uzaklıklarının Ortalaması

Bu isterin amacı kullanıcı 5 tuşuna bastığı zaman her bir dosya için sırası ile tüm noktaları bir dizi veya structa (yapıya) atayıp bu her bir noktayı birbiri ile karşılaştırıp, tüm noktaların birbiri ile olan uzaklıklarından yararlanarak tüm noktaların birbirine olan ortalamasını hesaplamaktır. Biz bu algoritma için projemizde dizi kullanmayı daha uygun ve efektif gördük. Bu yüzden de diziyi kullandık. Aynı şekilde bu isteri de sınadığımız zaman görüyoruz ki doğru sonuçları hızlı bir şekilde elde edebiliyoruz.

1. GERÇEKLEŞTİRİLEN TESTLER VE DEĞERLENDİRME

Yapmış olduğumuz bu projeyi birden fazla kişisel bilgisayarda çalıştırdığımızda ve test ettiğimizde aldığımız sonuç başarılı bir sonuç vermektedir. Fakat projenin güvenirliği için projenin ilgili kişilerce tekrardan tamamı ile test edilmesinde fayda vardır.

1. ÖZET VE SONUÇLAR

Bu projede belirtilen klasörde bulunan istenilen formatta olan dosyalardan 3 boyutlu nokta bilgileri çekilerek sonrasında yukarıda açıklanan işlemlere tabii tutulması amaçlanmaktadır. Yaptığımız bu proje bize C dili ile dosya okumanın nasıl yapıldığını, dosyaya verilerin nasıl kaydedildiğini bu verilere bir takım işlemlerin nasıl uygulanacağını göstermiştir. Aynı şekilde C dilinin günümüz nesnel dillere göre daha yorucu olmasına rağmen ne kadar güçlü bir dil olduğunu bir kez daha göstermiştir. Özetle, bu proje ile amaçlanan tüm işlemler sorunsuz bir şekilde programlanmış, test edilmiş ve bunun sonucunda başarılı olduğu görülmüştür.

##### Kaynakça

[1] <https://sozluk.gov.tr/?kelime=TASARIM>

[2] <https://www.reviversoft.com/tr/file-extensions/ascii>

[3] <https://www.techopedia.com/definition/938/binary-format>

[4] <http://www.two-sdg.demon.co.uk/curbralan/code/dirent/dirent.html>

