**ÜÇ BOYUTLU GÖRÜNTÜ DOSYALARI PROJESİ**

*Yağız Türer, Hüseyin Yılmaz*

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[yagizturer@gmail.com](mailto:yagizturer@gmail.com), [ylmzhsyn98@gmail.com](mailto:ylmzhsyn98@gmail.com)

***Özet***

Bu projede dinamik bellek yönetiminden faydalanarak matris yapılarıyla sayısı belirsiz dosyadan üç boyutlu noktaların verisini binary veya ascii olmak üzere belirtilen formatta okuma işlemi gerçekleştirildi ve bu okunan üç boyutlu nokta verileri bellekte saklanarak, bu noktalardan en yakın ve en yakın ve uzak olanları, bu noktaların mesafelerinin ortalaması, bu noktaları içine alacak en küçük küpün köşe koordinatları ve kullanıcıdan merkez ve yarıçap bilgisi alınan kürenin içinde kalan noktalar bulunarak bu sonuçlar çıktı dosyalasına kaydedildi.

**Giriş**

Çalışmada amaç binary ve ascii formatında veriler okuyarak, bu verileri struct yapılarında saklama ve üzerlerinde işlem yaparak işlem sonuçlarını tekrar uygun formatta kaydetmektir. Bizden sayısı belirsiz “.nkt” uzantılı dosyadan sayısı dosya başlığında belirtilmiş üç boyutlu noktaların koordinat ve varsa RGB renk değeri bilgilerini okumamız, bu bilgilerin hepsini bellekte aynı anda saklamamız, hepsi için sırasıyla başlık kontrolü gerçekleştirmemiz, en yakın ve en uzak noktaların koordinat ve indeks bilgilerini tespit etmemiz, bu noktaların hepsini içine alacak en küçük küpün tüm köşe koordinatlarını bulmamız, kullanıcından merkez koordinatları ve yarıçap uzunluğu tek bir kez istenen kürenin içinde bulunan tüm noktaları tespit etmemiz ve tüm noktaların kendi aralarındaki mesafelerinin ortalamasını hesaplamamız ve en sonunda tüm bu işlem sonuçlarını çıktı dosyasında kaydetmemiz istenmiştir. Bunun için öncelikle klasörde bulunan tüm “.nkt” uzantılı dosyaların isimleri tespit edilip bellekte saklandı. Her biri için ayrı ayrı başlık kontrolü ve nokta okuma işlemi yapıldı. Başlık kontrolü esnasında sürüm satırında “VERSION” stringi bulunup bulunmadığını tespit ettik. Alanlar satırında, nokta koordinatları

okunurken RGB renk bilgilerinin bulunup bulunmadığı bellekte saklandı. Eğer RGB bilgileri bulunması gerektiği halde herhangi bir satırda yalnızca koordinat bilgileri verilmişse veya RGB bilgileri bulunmaması gerekirken koordinatların yanına bu bilgi de eklenmişse hesaplama işlemleri yapılmadı. Daha sonra noktalar satırından dosyada bulunması gereken nokta sayısı bellekte kaydedildi ve dinamik bellek yönetimiyle noktaları içinde saklayacak yapıya uygun boyut atandı. Eğer noktaları okuma işlemi sırasında başlıkta verilenden farklı sayıda nokta tespit edildiyse uygun hata yazdırıldı ve diğer işlemler yapılmadı. Data satırında dosyanın binary formatında mı, yoksa ascii formatta mı açılacağı bilgisi alındı. Eğer noktalar binary formatında verildiyse farklı, ascii formatında verildiyse farklı bir okuma yöntemi kullanıldı. Nokta okuma işlemleri sırasında satırlardaki eksik bilgilerin kontrolü için çeşitli kontrollere yer verildi.

**Yöntem**

Klasörde bulunan dosyaların isimlerini almak için dirent.h kütüphanesini kullandık. Readdir fonksiyonu sayesinde sırasıyla dosyalarda gezildi ve son dört harfi “.nkt” olan dosyaların isimleri bir string dizisine kaydedildi. Nokta bilgilerini saklamak için xyz değerlerini tutacak üç double ve rgb bilgilerini tutacak üç integer tipinde değişkenden oluşan bir struct oluşturuldu. Her dosyanın nokta ve kontrol bilgilerini ayrı ayrı tutmak için ise bir dosyabilgi structı oluşturuldu. Bu struct dosyanın kontrol sonucunu, adını, rgb ve nokta sayısı dahil olmak üzere başlıktan okunan bilgileri, en yakın ve en uzak noktaların indislerini, noktaların kendi arasındaki mesafeyi ve tüm noktaları saklaması için noktastruct tipinde bir pointer içermektedir. Ayrıca noktaları tutan pointera bellekten yer ayrılırken malloc kullanıldığı için program sonlandırılırken bu dosyanın structında free işlemi gerekip gerekmeyeceğini tespit etmek için bir de freekontrol değişkeni ilave ettik.

Dosyanın ve noktaların kontrol işlemi birinci işlemde gerçekleştirilmektedir, önce başka bir işlem çalıştırılırsa bellekte olmayan veriler okunmaya çalışılacağı için her dosya için kontrol başta -1’e eşitlenmektedir. Diğer işlemler çalıştırıldığında bu kontrole bakılarak ilk işlemin çalışılıp çalışılmadığı tespit edilmekte ve eğer çalışmadıysa işlemlerin yapılmasının önüne geçilmektedir. Birinci işlem çalıştığı takdirde dosya uyumluysa kontrol 1’e, herhangi bir hata tespit edildiyse 0’a eşitlenmektedir. Tüm işlemlerin başında bu kontrol değerleri kontrol edilerek uyumsuz dosyalar için işlem yapılmasının önüne geçilmektedir. Başlık kontrolü ve dosya okuma işlemleri farklı fonksiyonlarda gerçekleştirildi.

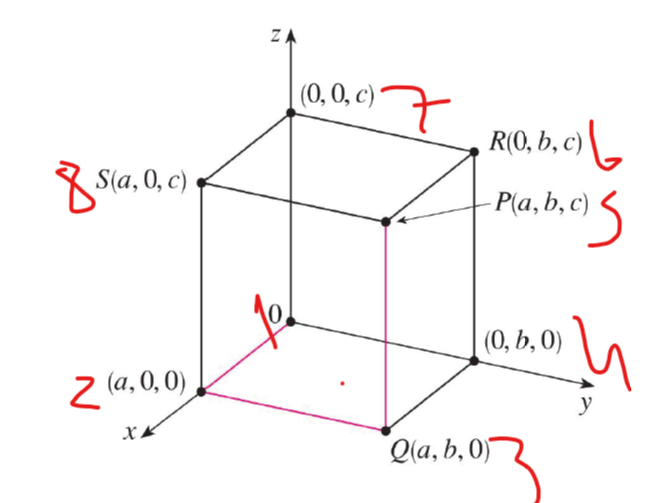
Binary ve ascii dosyalardan veri okumak için ayrı ayrı fonksiyonlar tanımladık ve farklı yöntemler kullandık. Dosyaların bu iki fonksiyondan birine gönderilmesi için önce baslikkontrol fonksiyonunda dosyanın kontrolünün 1 olarak atanmış olması gerekmektedir. Bu fonksiyonlara kontrolü 1 olarak giren dosyalar nokta bilgilerinde herhangi bir hata olması durumunda kontrolü 0 olarak döndürülebilmektedir. Ascii dosyalarda fgets fonksiyonuyla tek tek her satırı bütün olarak okuduktan sonra sscanf kullanarak bu stringi parçaladık. Daha sonra ilgili değişkenlere değer atanıp atanmadığını kontrol ederek uygunluğu kontrol ettik. Binary dosyalar ise satır satır okunmadığı için istenilen byte ile okuma gerçekleştiren fread fonksiyonunu kullandık. RGB bilgileri yoksa her seferinde 3 float boyutunda, RGB bilgileri varsa da yanında ayrıca 3 int boyutunda byte okumaya ayarladık. Herhangi bir satırda rgb eksiği veya fazlası varsa okunan nokta sayısı farklı çıkmakta ve hata yazdırılmaktadır.

2. ve 5. işlemler için tek bir fonksiyon tanımlandı. Sırasıyla tüm noktaların arasındaki mesafe bulunurken hem en yakın ve en uzak noktalar tespit edildi, hem de ortalamaları hesaplandı. Böylece iki işlem için aynı döngü iki defa çalıştırılmadı ve programın çalışma süresi kısaltıldı. Bu fonksiyonun daha önce çalışıp çalışılmadığını tespit etmek için ise başta tüm ortalama değerleri -1 olarak atanmıştır. Mesafelerin ortalaması negatif bir değer olamayacağı için, hesaplama yapılıp yapılmayacağına bu değer karar vermektedir. Bu değer değişmişse doğrudan yazdırma işlemine geçilmektedir. Bu fonksiyon ortalama ve en yakın-en uzak noktaların indis bilgilerini bellekte saklayacağı için struct dosyabilgi tipinde tanımlanmıştır.

Noktalar arası mesafeler hesaplanırken yazı karmaşasından kaçınmak için ayrı bir fonksiyon tanımlamayı tercih ettik. Bu mesafe hesaplanırken kare ve karekök bulmak için math.h kütüphanesinden faydalandık. Fonksiyon iki nokta için x,y ve z değerleri almakta ve aralarındaki mesafeyi döndürmektedir.

Kürenin içinde kalan noktalar tespit edilirken her seferinde koordinat ve yarıçap bilgileri istenmemesi için bu bilgiler menü fonksiyonunda işlem çağırıldıktan sonra isteniyor ve her dosyayla birlikte fonksiyona yollanıyor. Tek tek her noktanın merkezle arasındaki mesafe hesaplanarak bu değer yarıçaptan küçük veya eşit olanlar çıktı dosyasına kaydediliyor. Bu fonksiyon struct üzerinde herhangi bir değişiklik yapmadığı için void tipinde tanımlanmıştır.

Tüm noktaları içine alacak en küçük küpü yazdırmak için en küçük x, y ve z değerleri tespit edilerek bulunan kenar uzunluğunu köşelere eklenmiştir. Kenar uzunluğu ise en büyük değerlerden en küçük değerlerin farkları arasından en büyüğü bulunarak tespit edilmiştir. Küpün köşelerini yazdırırken aşağıdaki görselden faydalanılmış ve kenar uzunluğu bu sırayla eklenmiştir:



Küp fonksiyonu da dosya structında herhangi bir değişiklik yapmadığı için void tipinde tanımlandı.

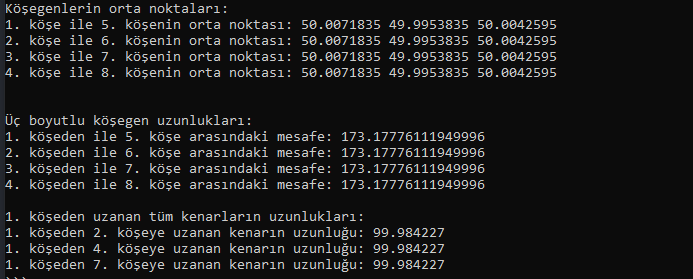
Eğer ki menüden 0 seçilirse programın düzgün biçimde sonlanması için freekontrolü 1 olan tüm nokta structları için free işlemi uygulanmaktadır.

Menü yazdırma ve seçim işlemleri için void tipinde bir fonksiyon tanımladık. Bu fonksiyon 0. işlem seçilene dek sürmektedir. Bu fonksiyon bittikten sonra main fonksiyonunda başka bir işlem olmadığı için program hatasız şekilde sonlanmaktadır.

**Sonuçlar**

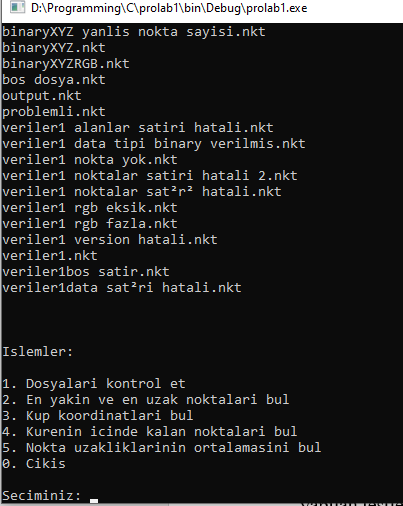
Tüm bu işlemler sonucunda klasörde bulunan tüm dosyalardan bilgi ve üç boyutlu nokta okuma işlemleri başarıyla gerçekleştirildi. Dosyalarda olabilecek hatalar için çeşitli kontroller konularak runtime hatalarının önüne geçilmeye çalışıldı. Program tüm dosyalar için gerekli hesaplamaları yaparak çıktıları aynı klasörde bulunan bir “output.nkt” dosyasına kaydetmektedir. Bu dosya daha önceden varsa 1. işlem her çalıştırıldığında sıfırlanmakta, yoksa oluşturulmaktadır.

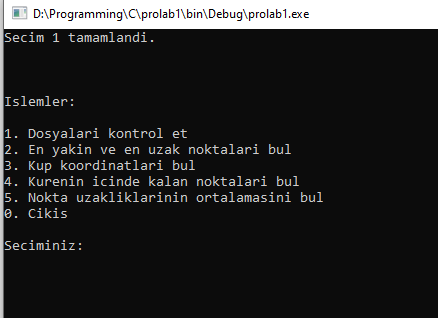
Küp koordinatlarının doğruluğunu test etmek için köşegenlerin uzunluklarını ve orta noktalarını kıyasladık, buna ek olarak bir köşeden çıkan tüm kenarların uzunluklarını kıyaslayarak yazdırılan koordinatların bir küp oluşturduğunu teyit ettik. Veriler1.nkt dosyası için yapılan testlerin sonuçları aşağıdaki görselde verilmiştir:

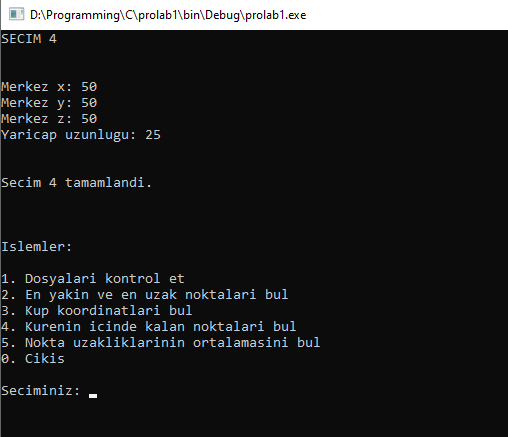


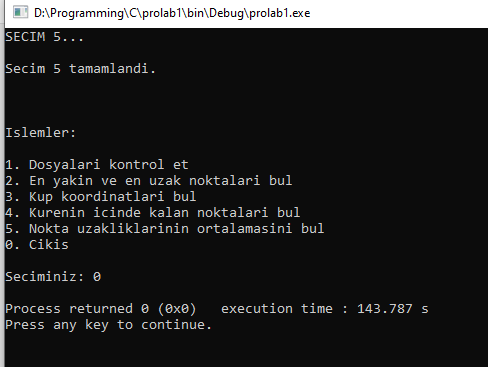
Program codeblocks ile derlenerek Windows 10 işletim sisteminde, gcc ile Linux terminalinden derlenerek de Ubuntu 19.04 kurulu bir uzak sunucuda test edilmiş, yapılan testlerde herhangi bir runtime hatasıyla karşılaşılmamıştır.

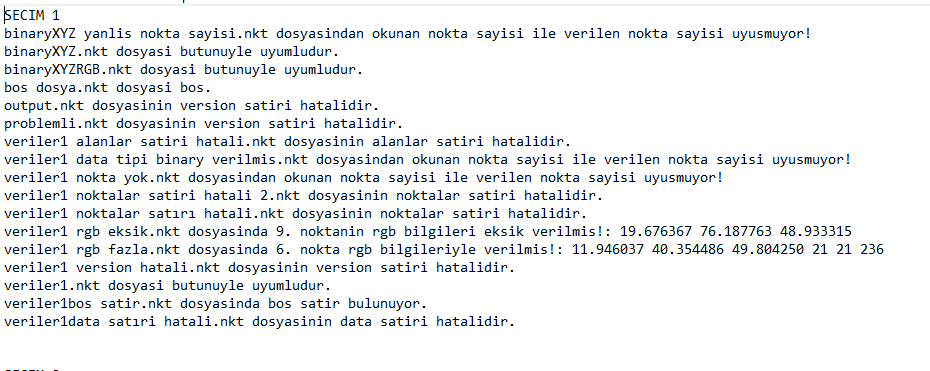
**Çıktılar**

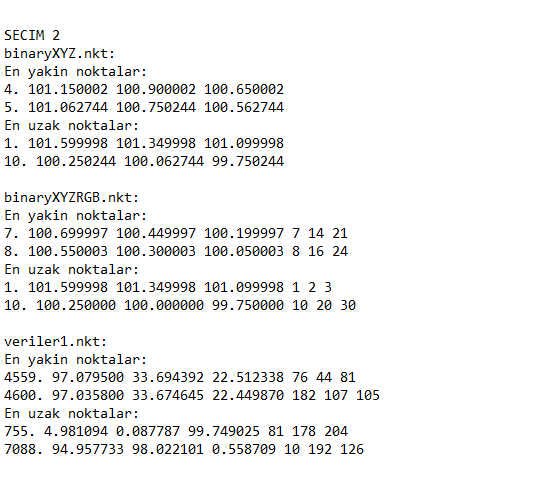
****

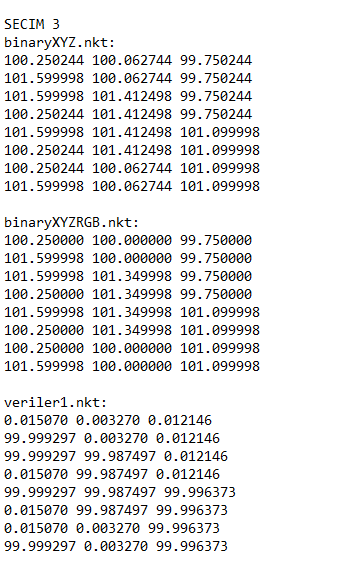


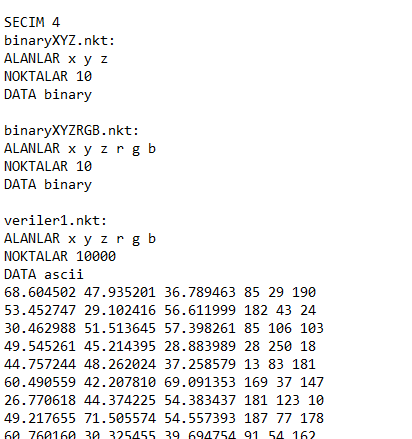


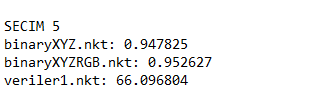
****



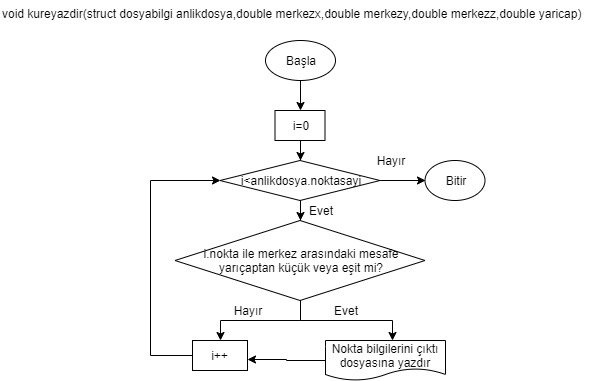
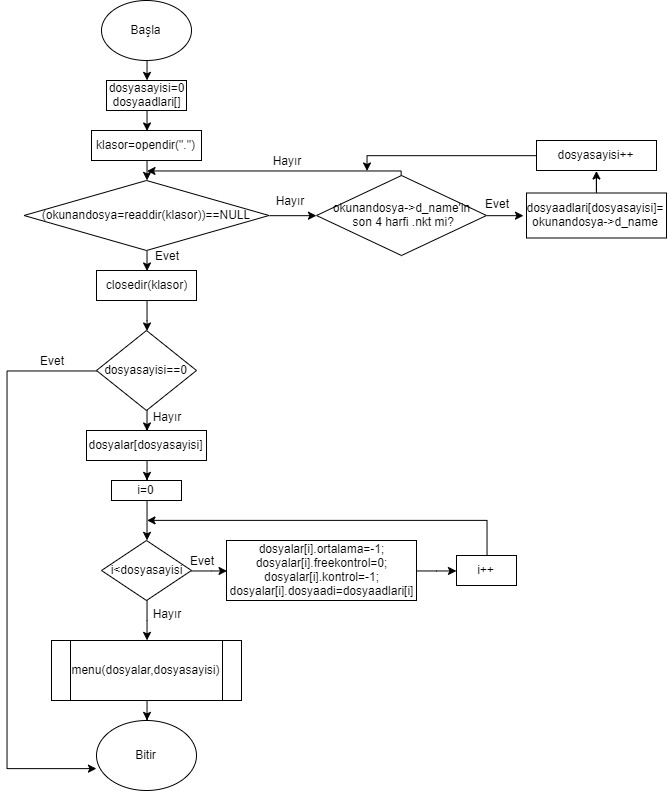


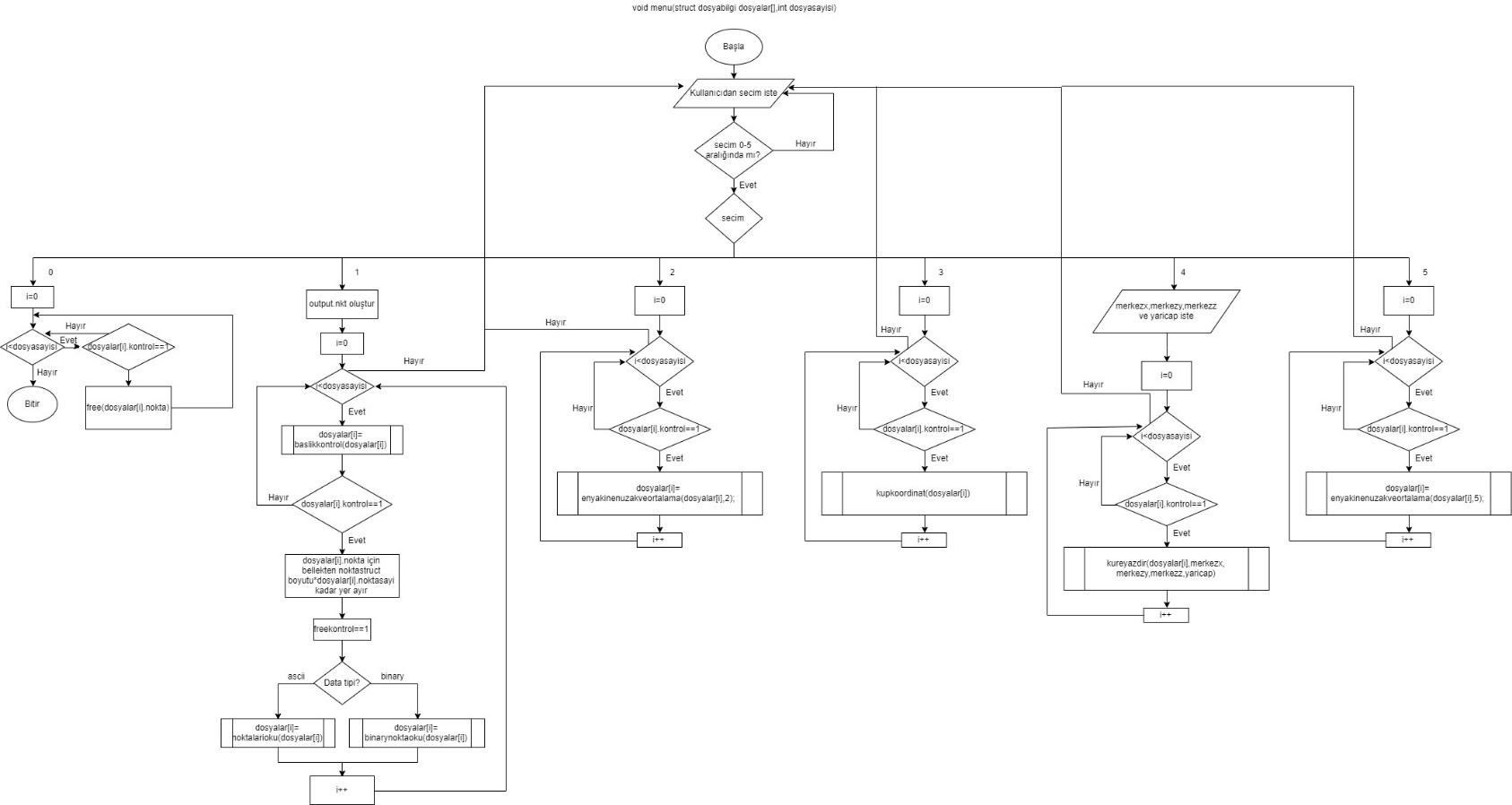


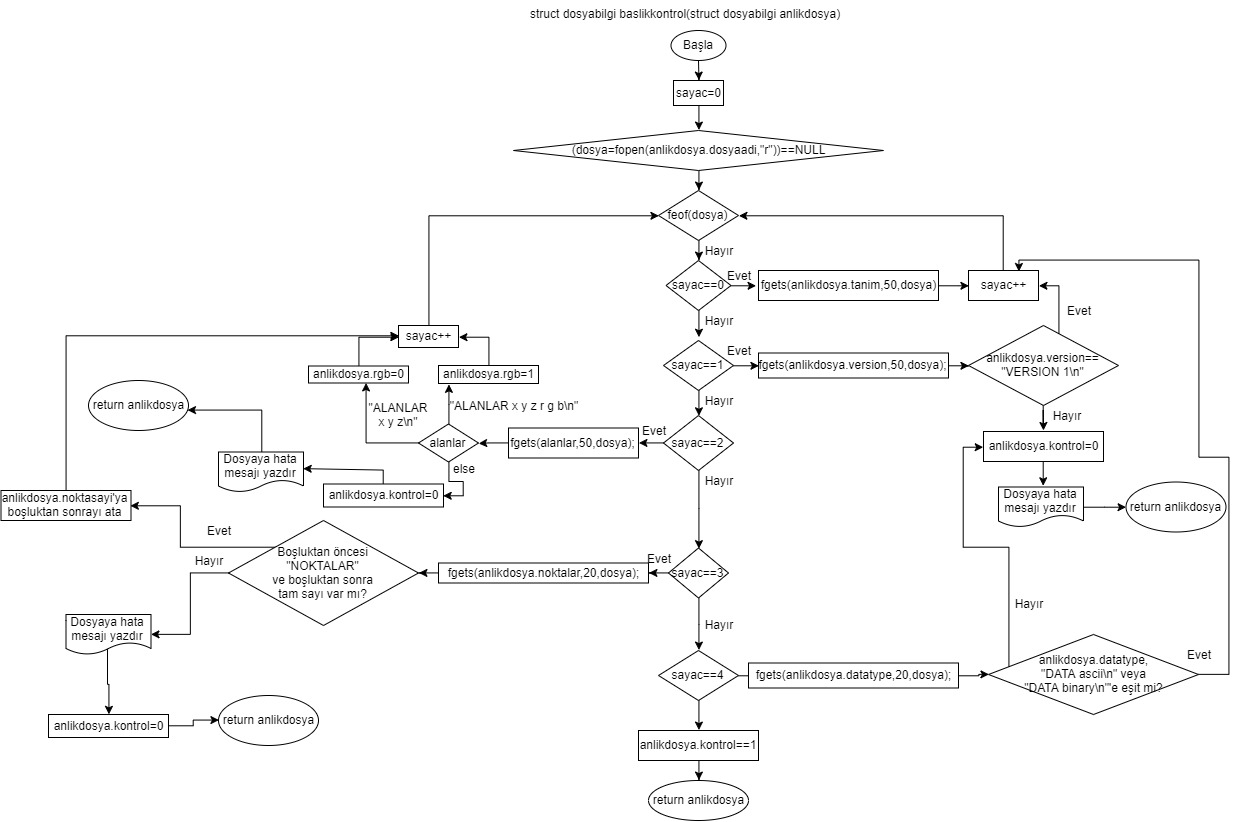


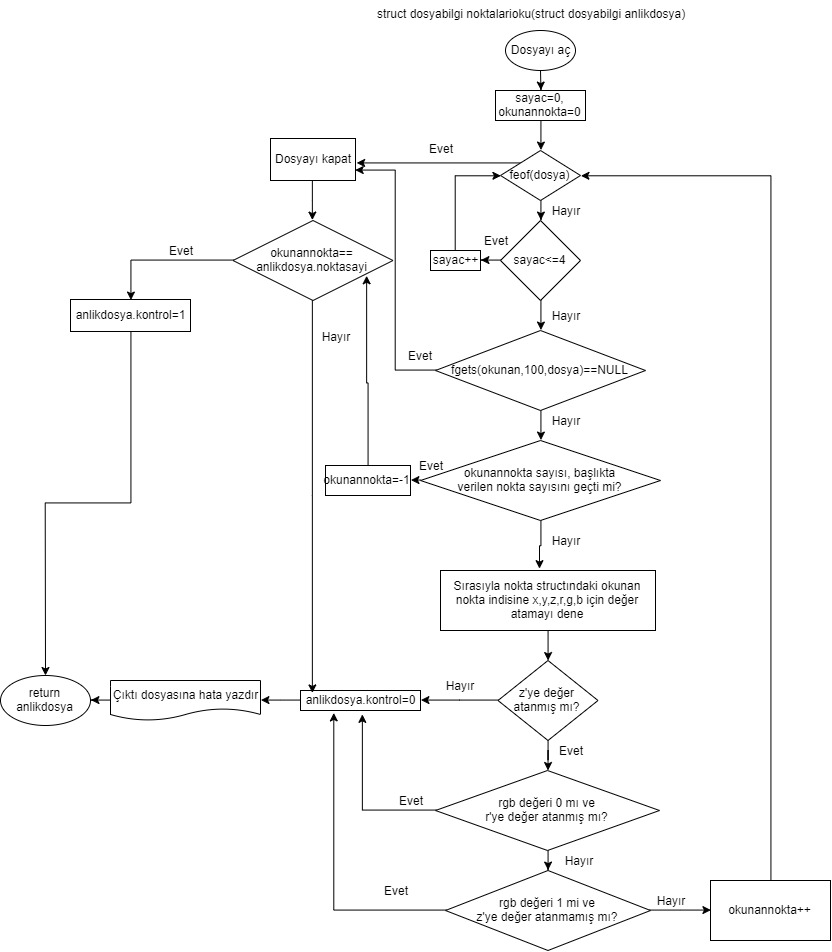


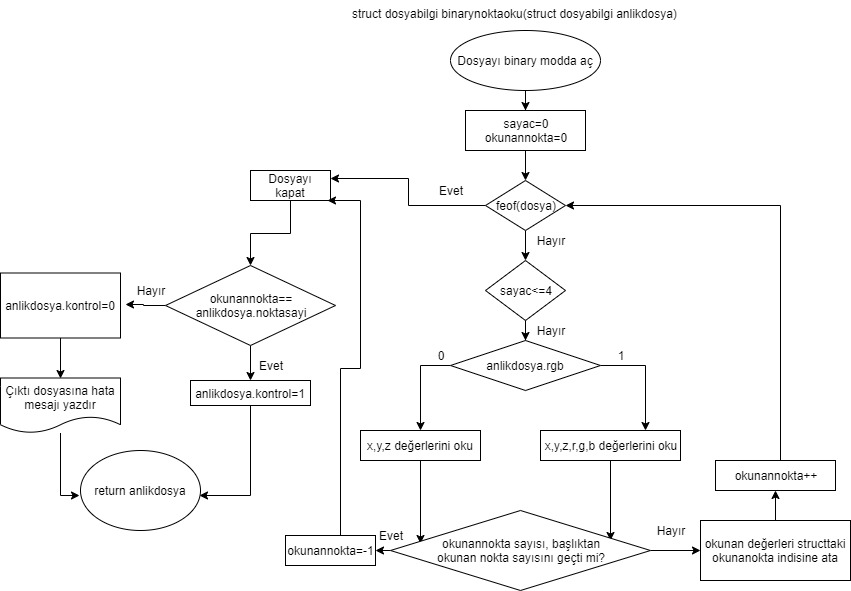
**Akış Şemaları:**

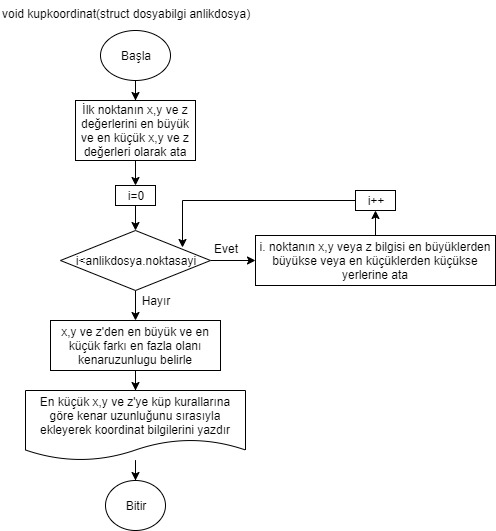
****

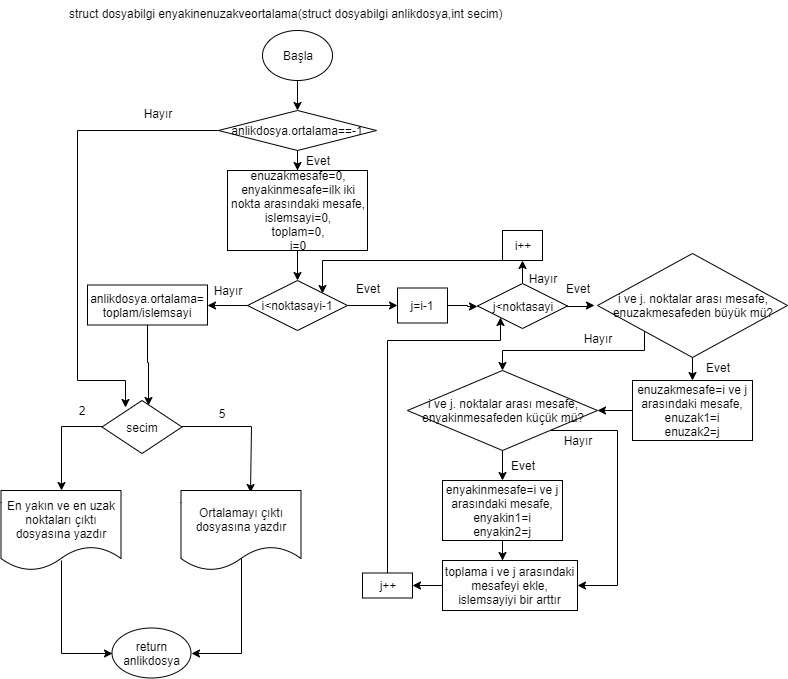
****

****

****

****

****

****

Şema çizimleri draw.io üzerinden yapılmıştır.

**Kaynakça:**

<https://www.tutorialspoint.com/c_standard_library/c_function_fread.htm>

<https://www.tutorialspoint.com/c_standard_library/c_function_sscanf.htm>

<https://pubs.opengroup.org/onlinepubs/007908799/xsh/dirent.h.html>

<https://www.engineeringtoolbox.com/distance-relationship-between-two-points-d_1854.html>

<https://www.programiz.com/c-programming/c-structure-function>

<https://www.programiz.com/c-programming/c-dynamic-memory-allocation>

<https://www.tutorialspoint.com/c_standard_library/string_h.htm>

<https://fresh2refresh.com/c-programming/c-file-handling/>

<https://stackoverflow.com/questions/2347770/how-do-you-clear-the-console-screen-in-c>

Onur Gök ve Suhap Şahin 2018-2019 Programlama Dersi Slaytları