**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ** **LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**YAZILIM MÜHENDİSLİĞİ PROGRAMI**

**ÖDEV KONUSU**

**PROGRAMLAMA LAB 1 DERSİ ÖDEV 1**

**CONSOLE MENU**

**Hazırlayanlar**

**Gürel BİLGİN**

**220502041**

**https://github.com/GurelBilgin**

**Berkay ARAS**

**220501033**

**https://github.com/brkyaras**

**DERS SORUMLUSU**

**DR. ÖĞR. ÜYESİ Nur Banu ALBAYRAK**

**30.10.2023**

**İÇİNDEKİLER**

[1. ÖZET (ABSTRACT) 3](#_Toc136267779)

[2. GİRİŞ (INTRODUCTION) 3](#_Toc136267780)

[3. YÖNTEM (METHOD) 3](#_Toc136267781)

[3.1 Algoritma ve Fonksiyonlar 3](#_Toc136267782)

[3.2 Kullanılan Teknolojiler ve Kaynaklar 3](#_Toc136267783)

[4. SONUÇ VE ÖĞRENİLEN DERSLER 3](#_Toc136267784)

[5. KAYNAKÇA 3](#_Toc136267785)

# ÖZET

Bu ödevde, Python programlama dili kullanılarak bir "console menu" oluşturuldu. Bu console menu, çeşitli matematiksel ve algoritmik işlemleri gerçekleştirmemize olanak tanır. Öğrenci, bu işlevleri özyinelemeli fonksiyonlar kullanarak geliştirdi.

Projede Python dilinin çeşitli yeteneklerinden yararlanıldı. Temel liste işlemleri, dizi işlemleri, metin işleme ve özyinelemeli işlevler gibi kavramlar ele alındı. Aynı zamanda hata yönetimi ve veri türü dönüşümü gibi konular içeriğe dahil edildi.

Projede şunlar yer alır:

1. k'nıncı En Küçük Elemanı Bulma

2. En Yakın Çifti Bulma

3. Bir Listenin Tekrar Eden Elemanlarını Bulma

4. Matris Çarpımı

5. Bir Text Dosyasındaki Kelimelerin Frekansını Bulma

6. Liste İçinde En Küçük Değeri Bulma

7. Karekök Fonksiyonu

8. En Büyük Ortak Bölen Bulma

9. Asallık Testi

10. Daha Hızlı Fibonacci Hesabı

Projenin bazı kısımları hala geliştirilme aşamasındadır ve bazı zorluklarla karşılaşıldı. Ancak bu ödev, Python programlama yeteneklerini geliştirmek için harika bir fırsat sağladı. Öğrenci, öğrenme deneyimi kazandı ve projenin gelecekteki geliştirme potansiyelini görmekten heyecan duydu.

# GİRİŞ

Bu projenin temel amacı, Python programlama dilini kullanarak çeşitli matematiksel görevleri yerine getirebilen özelleştirilmiş fonksiyonlar geliştirmektir. Projede kullanılan fonksiyonlar, kullanıcıya k'ıncı en küçük elemanı bulma, iki sayının toplamının belirli bir hedefe en yakın çiftini bulma, bir listenin tekrar eden elemanlarını bulma, matris çarpımı yapma, bir metin dosyasındaki kelime frekanslarını hesaplama, bir listedeki en küçük değeri bulma, bir sayının karekökünü hesaplama, iki sayının en büyük ortak bölenini bulma ve bir sayının asal olup olmadığını kontrol etme gibi farklı matematiksel işlemleri gerçekleştirmektedir.

Projede kullanılan ana yöntem, Python programlama dili ve özyinelemeli fonksiyonlar kullanmaktır. Bu fonksiyonlar, işlemlerin tekrarlı ve veriye dayalı olarak nasıl gerçekleştirileceğini göstermektedir. Projede ayrıca, kullanıcının metin dosyalarından veri okuma ve işleme yeteneklerini geliştirmek için Python'un dosya işleme yetenekleri kullanılmıştır.

Bu çalışmanın sonunda, Python programlama dilini kullanarak karmaşık matematiksel işlemleri nasıl çözebileceğinizi ve özelleştirilmiş fonksiyonlar oluşturarak bu işlemleri nasıl otomatize edebileceğinizi öğrenmiş olacaksınız. Ayrıca, özyinelemeli fonksiyonlar, algoritmik düşünme ve problemleri nasıl parçalayıp çözeceğinizi anlama konularında size yardımcı olacaktır.

# YÖNTEM

Bu bölümde, projede benimsenen ana metot, takip edilen yöntemler ve kullanılan araçlar hakkında kapsamlı bir açıklama sunulmaktadır. Bu bölüm, projenizin temel yapısını ve nasıl işlediğini daha iyi anlamak için okuyucuya bir bakış sunar.

**3.1. Algoritma ve Fonksiyonlar**

Proje, kullanıcıya farklı matematiksel ve hesaplama görevlerini gerçekleştirmek için bir dizi özelleştirilmiş Python fonksiyonu sunar. Bu fonksiyonlar aşağıdaki gibidir:

* **k\_kucuk(k, liste):** Bu fonksiyon, verilen bir listedeki k. en küçük elemanı bulur. Kullanıcı, k ve bir liste girerek işlemi gerçekleştirir.
* **en\_yakin\_cift(hedef, liste):** Bu fonksiyon, bir listedeki sayıları toplayarak belirtilen bir hedefe en yakın olan sayı çiftini bulur. Kullanıcı, hedefi ve bir liste girerek işlemi gerçekleştirir.
* **tekrar\_eden\_elemanlar(liste):** Bu fonksiyon, bir listede tekrar eden elemanları bulur ve kullanıcıya geri döner.
* **matris\_carpimi(matris1, matris2):** Bu fonksiyon, iki matrisin çarpımını bulur. Kullanıcı, iki matrisi girdi olarak vererek işlemi gerçekleştirir.
* **kelime\_frekansi(text\_dosya\_yolu):** Bu fonksiyon, bir metin dosyasındaki kelimelerin frekansını hesaplar. Kullanıcı, metin dosyasının yolunu girerek işlemi gerçekleştirir.
* **en\_kucuk\_degeri\_bul():** Bu fonksiyon, kullanıcının girdiği bir listedeki en küçük değeri bulur.
* **karekok\_hesapla(N, x0, tol=1e-10, maxiter=10):** Bu fonksiyon, verilen bir sayının karekökünü hesaplar. Kullanıcı, sayıyı ve başlangıç tahminini girerek işlemi gerçekleştirir.
* **eb\_ortak\_bolen(sayi1, sayi2):** Bu fonksiyon, iki sayının en büyük ortak bölenini hesaplar. Kullanıcı, iki sayıyı girerek işlemi gerçekleştirir.
* **asal\_veya\_degil(sayi):** Bu fonksiyon, verilen bir sayının asal olup olmadığını kontrol eder. Kullanıcı, bir sayı girerek işlemi gerçekleştirir.
* **hizlandirilmis\_fibonacci(n):** Bu fonksiyon, verilen bir sıradaki Fibonacci sayısını hesaplar. Kullanıcı, sıra numarasını girerek işlemi gerçekleştirir.

**3.2. Kullanılan Teknolojiler ve Kaynaklar**

Bu projede Python programlama dili kullanılmıştır. Projede ayrıca farklı matematiksel hesaplamaları gerçekleştirmek için özel fonksiyonlar oluşturulmuştur. Ayrıca, metin dosyasından veri okuma ve işleme için Python'un standart kitaplığı kullanılmıştır.

Projede kullanılan temel kaynaklar, Python belgeleri, kaynak kodları ve öğreticileridir. Ayrıca, hesaplama algoritmaları ve veri işleme süreçleri üzerine araştırma ve okuma yapılmıştır.

Bu yöntemler ve araçlar, projenin başarılı bir şekilde gerçekleştirilmesini sağlamıştır.

# SONUÇ VE ÖĞRENİLEN DERSLER

Bu ödevin tamamlanması sonucunda, Python programlama dilini kullanarak farklı matematiksel problemleri çözme becerilerimizi önemli ölçüde geliştirdik. Özellikle, özyinelemeli fonksiyonları kullanarak karmaşık hesaplamaları nasıl gerçekleştireceğimizi öğrenmek, matematiksel ve algoritmik düşünme yetilerimizi artırmamıza yardımcı oldu.

Öğrendiğimiz en önemli derslerden biri, problemleri daha küçük ve daha anlaşılabilir parçalara bölmek ve her bir parçayı çözmek için özelleştirilmiş fonksiyonlar geliştirmektir. Özyinelemeli fonksiyonlar, bu tür işlemleri gerçekleştirmek için güçlü bir araçtır ve büyük ve karmaşık sorunları daha yönetilebilir hale getirir. Özellikle Fibonacci dizisinin hesaplanması gibi özyinelemeli problemlerde bu konsepti kullanarak daha hızlı sonuçlar alabiliriz.

Ayrıca, projede Python'un dosya işleme yeteneklerini kullanarak metin dosyalarından veri okuma ve işleme becerilerini geliştirdik. Bu, gerçek dünya verileri üzerinde çalışırken önemli bir yetenektir.

Bu projenin sonuçları, özellikle matematiksel ve hesaplamalı problemlerle uğraşan kişiler için faydalıdır. Ayrıca, programlama becerilerini geliştirmek isteyen herkes için bu tür özelleştirilmiş fonksiyonların nasıl oluşturulacağına dair fikirler sunabilir.

Sonuç olarak, bu ödev, Python programlama dilini kullanarak matematiksel sorunları çözme yetilerini geliştirmek ve özyinelemeli fonksiyonları anlamak için değerli bir deneyimdi. Ayrıca, karmaşık problemleri parçalara ayırma ve çözme konusunda yeni beceriler kazandık. Bu projenin sonuçları, gelecekteki programlama çalışmaları ve matematiksel problemlerle başa çıkmak için önemli bir temel oluşturacaktır.

# KAYNAKÇA

* Python.org. (2023). Python Documentation. https://docs.python.org/3/
* Python Sıfırdan İleri Seviye Programlama. (2020). Udemy. Mustafa Murat Coşkun

https://www.udemy.com/course/sifirdan-ileri-seviyeye-python/

* Sıfırdan İleri Seviye Python Programlama. (tarih bilinmiyor). BTK Akademi. Sadık Turan
* Project Euler. (tarih bilinmiyor). Erişildiği tarih: [30.10.2023]. URL: [https://projecteuler.net/archives]