**T.C**

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

**Bilgisayar Mühendisliği Programı**

**Console Menu**

**Hazırlayan**

**Anıl Erdoğan- 220501006**

**GitHub Link: https://github.com/Kwalmm**

**AKIN TURAN-** **220501013**

**GitHub Link: https://github.com/coldwraith44**

**DERS SORUMLUSU**

**ÖĞR. GÖR. ŞEVVAL ŞOLPAN**

**ÖĞR. GÖR. ERAY DURSUN**

**29.10.2023**

**İÇİNDEKİLER**

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

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

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

[3.1 Örnek Alt Başlık 3](#_Toc136267782)

[3.2 Örnek Alt Başlık 3](#_Toc136267783)

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

[5. KAYNAKÇA 3](#_Toc136267785)

# ÖZET

Bu çalışmada istenilen “Console Menu” yapısı recursive olan veya olmayan fonksiyonlar yardımı ile oluşturuldu ve temel menü yapısı terminale yansıtıldı ayrıca gui için tkinter kütüphanesi kullanıldı

# GİRİŞ

Bu çalışmanın temel amacı oluşturulan “Console Menu” ile kullanıcıya başarılı bir deneyim sunmak. Yazılan fonksiyonlar sayesinde kodun bakımı ve ilerde gelebilecek güncellemelerin uygulanması kolaylaştırıldı. Ayrıca yazılan fonksiyonlar sayesinde kodun modülerliği ve yeniden kullanılabilirliği arttırıldı.

# YÖNTEM

Yöntem kısmında her soruya ve menüye tek tek nasıl yapıldığına değineceğiz;

## 3.1) Bu soruda istenilen k. küçük elemanı bulmak için basit bir metot kullanıldı öncelikle verilen liste sıralandı ardından liste uzunluğu tek tek kontrol edildi ve geçersiz veya eksik değerler için geri dönüş sağlandı son olarak listeye verilen indexin bir eksiği girildi ve doğru sonuca ulaşılması sağlandı

**3.2)** Bu soruda öncelikle fark ve o iki sayı için rastgele değerler girildi ardından for döngüsü ile önce listenin elemanlarına erişildi ardından o listenin kopyası oluşturulup oluşan kopya listeden erişilen eleman çıkarıldı ve kalan liste yeni bir döngü ile tek tek erişilen ilk eleman ile toplanıp kontrol edilmeye başlandı gerekli if yapısı ile büyük küçük kontrolü yapıldı ve son olarak doğru sonuç elde edildi

**3.3)** Bu soruda öncelikle boş bir liste oluşturuldu ardından yapılan basit bir list comprehension metodunun içindeki if yapısı ile tek tek kontrol edilerek doğru sonuca ulaşıldı en son cevap için ise slicing metodu kullanıldı

**3.4)** 1. Aşama: 2 adet liste oluşturuyoruz. Bunun sebebi iki listeyi birleştirip bir matris formu oluşturmak.  
 2. Aşama: Listelerden biri büyük liste, diğeri ise küçük bir listedir. Büyük listenin içerisine küçük listeler yerleştirilecek ve matris formunda bir liste oluşacaktır. Listeleri büyük ve küçük diye ayırmak için iç içe "for" döngüsü kuruyoruz.  
 3. Aşama: Bizim amacımız bir matris çarpması yapmak olduğundan 2. listenin "zip" formunu alıyoruz ve "for" dönügüsüne katıyoruz. Böylece sırayla sütunların elemanlarınıda elde ediyoruz ve matris çarpmasını matematiksel olarak gerçekleştiriyoruz. Bunun için "sum" fonksiyonunu ve "zip(i,i2)" ile satır ile sütunun eşleştirilmiş halini elde edip çarpıp toplayarak sırayla yeni matrisin elemanlarını oluşturuyoruz. Elemanlar küçük liste olarak kaydedilip satırları oluşturuyor satırlarda sütunları oluşturuyor  
 4. Aşama final: Elde ettiğimiz sonuçları önce bir liste yapıp bu listeyi büyük listeye ekleyerek matrisin satırlarını oluşturduk. En son elde ettiğimiz sonucu return ile döndürüyoruz.

**3.5)** 1. aşama: "with open(dosya\_adi,'r') as d: kelimeler = d.read().split()" ile dosyayı açıp okuma modunda okuyoruz. Okunan dosyanın elemanları bir liste yapıyoruz bunun için "split()" kullanıyoruz.  
 2. aşama: "def kucukharf(kelime): return kelime.lower()" ile bu kelimeleri küçük harfe çeviriyoruz ki ileride eşleşme sorunu kodda gerçekleşmesin. Bu esnada "map" fonksiyonundanda faydalanıyoruz.  
 3. aşama: "def sayac(kelime\_sayac,kelime):" Bize bir kelimenin kaç kez kullanıldığını bir sözlükte tutuyor. Eğer kelime varsa 1 yoksa 0 değerini döndürüyor.  
 4. aşama: elde ettiğimiz sayısal verileri toplamak için "kelime\_say = reduce(sayac,kelimeler,{})" satırını kullanıyoruz ve {} ile bir listede bunları birleştiriyoruz  
 5. aşama final: "return kelime\_say" ile en son elde ettiğimiz listeyi döndürüyoruz. Bu kodu kullanırken dosya adını siz formata uygun biçimde yazmalısınız.

**3.6)**  Bu soruda öncelikle liste alınır ardından liste üzerinde işlem yapmak için bir onksiyon oluşturuldu. İlk olarak uzunluğu if yapısı ile kontrol edildi ardından başka bir if yapısı kontrol ederek küçük sayıyı bulmaya yarayan başarılı bir recursivce fonksiyon elde edilmiş oldu

**3.7)** 1. aşama: "def karekok()" fonksiyonu için 4 parametre atıyoruz. "sayi" parametresi karekökünü bulacağımız sayıyı almaktadır. "tahmin" parametresi bu karekök değerinin bir insan tarafından yapılan tahmini değeridir. "maxiter" parametresi deneme sayısı yerine geçer kullanacağımız formülde belli başlı işlemler gerçekleşeceğinden bu işlemlerin yapılma sayısı bu parametrede belirlenir. "x" kontrol sayacı ise maxiter ile kıyaslanarak işlemler ona göre kontrollü gerçekleşir.  
  
 2. aşama: "tol" değişkeni hata payımızı simgeler. Karekök olarak en son bulacağımız ilkel değer "tol" hata payı miktarı ile kıyaslanır. Hata payı "tol"un altındaysa işlem sonlandırılır.  
  
 3. aşama: "maxiter" kontrol çubuğu göreve başlar ve "y = (1/2)\*(tahmin + (sayi/tahmin))" formülü ilk kez uygulanır. Makinaya ait ilk karekök tahmini bulunur. Hata payı hala fazlaysa ve deneme sayısı aşılmamışsa işlem devam eder burda "else:" çubuğuna girer ve özyinelemeli bir fonksiyon elde etmiş oluruz. Çünkü burada fonksiyon kendini bir daha çağırır. Bu esnada "return karekok(sayi, y, maxiter, x+1)" formülün kuralı gereği "tahmin" parametresi "y"ye atanır ve "x" 1 artar. Baştaki "y = tahmin" kod bloğu bizim için önemlidir. Fonksiyon kendini gene çağırdığında "y"nin değeri kaybolmamalıdır.  
  
 4. aşama final: Artık nihayetinde hata payıda azalmaya başlar ve deneme sayısıda yeterliyse istenilen sonuç döndürülür. Eğer hata payı hala istenilenden fazla ise ve deneme sayısıda kalmamışsa kodun son bloğundaki "return "Yeterli denemede sayıya ulaşılmadı. Karekök değeri olarak hesaplandı:", y" döndürülür.

**3.8)** Bu soruda kullanıcıdan alınan iki integer sayının işlenmesi için yine recursive bir fonksiyon kullanılır. Önce bir kontrol yapısı ile 0 olup olmadığı kontrol edilir sorun çıkmazsa alınan iki sayıdan diğerine bölünür ve alınan integer değer 0 olana kadar devam eder

**3.9)**  Bu soruda kullanıcıdan integer değer alınır ve asal olup olmadığını kontrol etmek için recursive bir fonksiyon oluşturulur fonksyion 2 değer alır ve kullanıcıdan alınmayan değere başlangıç olarak 2 verilir öncelikle yine bir if yapısıyla eşit olma durumları kontrol edilir ve eğer eşitse true döndürülür eğer değilse sayılar farklıdır ve bölünür tam bölünüyorsa asal olamaz ve false döndürülür eğer tam değilse fonksiyon kendini tekrar çağırır ama bu sefer sabit olan değer bir arttırılır işlem girilen sayı arttırılana eşit olana kadar devam eder eğer asal çıkmazsa True döndürülür.

**3.10)** 1. aşama: 4 Adet parametre istek doğrultusunda n,k,fibk ve fibk1 koda eklenmiştir. "n" kodun duracağı yer, "k" sıra numarası ve "fibk" ile "fibk1" fibonacci listelerini alt alta oluşturur.  
 2. aşama: "if k == n:" bloğu kodun uygun anda durmasını sağlar. "Return" tek başına fonksiyonu bitirir. Kod özyinelemeli olduğundan devam ettiği anlarda "else:" bloğuna girer ve "fibo\_dizisi(n, k+1, fibk+fibk1, fibk)" satırı ile fibonacci dizisi sürdürülür.  
 3. aşama final: son olarak kullanıcı bir sınır sayısı verir ve kod çalışır.

**3.11)**  Kurulan while ile birlikte menüden çıkılması veya yanlış değer girişinin belirtilmesi sağlandı

**3.Menu)** Menüyü tasarlarken sade ve kullanıcının cevaplarıyla etkileşimli bir menü kuruldu kullanıcıdan alınan inputlar inte floata veya for döngüleri ile listeye çevrilerek yukarıda açıklanan fonksiyonlar ile etkileşime girmesi sağlandı.

# SONUÇ VE ÖĞRENİLEN DERSLER

Ödevin sonucunda fonksiyonlar ile birlikte bir gui aracılığıyla oluşturulmuş başarılı bir console menu elde ettik. Farklı kütüphane kullanımları ile recursive fonksiyonlar hakkında olan bakış açımız arttı.

**5.KODUN SON HALİ**

#Console Menu

#Anıl Erdoğan Bilgisayar Mühendisliği 220501006

#Akın Turan Bilgisayar Mühendisliği 220501013

import tkinter as tk

from functools import reduce

import os

import math

def k\_kucuk(k:int,l:list):

l.sort()

if k>len(l):

print("Girdiğiniz değer listenin eleman sayısını aşıyor.")

print("Tekrar deneyiniz!")

elif k==0 or k<0:

print("Geçersiz değer girdiniz.")

print("Tekrar deneyiniz!")

else:

sonuc=l[k-1]

return sonuc

def en\_yakin\_cift(x:int,y:list):

fark=10

ilk\_Sayi=5

ikinci\_sayi=5

for i in range(len(y)):

yeni\_liste=y.copy()

yeni\_liste.pop(i)

for j in range(len(yeni\_liste)):

z=y[i]+yeni\_liste[j]

if abs(x-z)<fark:

fark=abs(x-z)

ilk\_Sayi=y[i]

ikinci\_sayi=yeni\_liste[j]

return [ilk\_Sayi,ikinci\_sayi]

def tekrar\_eden\_elemanlar(x:list):

tekrar\_eden\_elemanlar=[]

[tekrar\_eden\_elemanlar.append(i) for i in x if x.count(i)>1 and i not in tekrar\_eden\_elemanlar]

return tekrar\_eden\_elemanlar[:]

def matris(liste1,liste2):

sonuc = []

for i in liste1:

sonuc2 = []

for i2 in zip(\*liste2):

sonuc2.append(sum(a\*b for a,b in zip(i,i2))) #[[1,2,3], [4,5,6]] ve [[7,9,11],[8,10,12]

sonuc.append(sonuc2)

return sonuc

def kelime\_sayisi(dosya\_adi):

with open(dosya\_adi,'r') as d:

kelimeler = d.read().split()

"""txt\_directory ="C:/Users/PC/Desktop/Python Fell"

txt\_filename = "plabtext.txt"

txt\_path = os.path.join(txt\_directory, txt\_filename)"""

def kucukharf(kelime):

return kelime.lower()

kelimeler = list(map(kucukharf,kelimeler))

def sayac(kelime\_sayac,kelime):

kelime\_sayac[kelime] = kelime\_sayac.get(kelime,0)+1

return kelime\_sayac

kelime\_say = reduce(sayac,kelimeler,{})

return kelime\_say

def en\_kucuk\_deger(x:list):

if len(x)==1:

return x[0]

else:

if x[0] < x[1]:

return en\_kucuk\_deger([x[0]]+x[2:])

else:

return en\_kucuk\_deger([x[1]]+x[2:])

def karekok(sayi, tahmin, maxiter, x=0):

tol = 10\*\*(-10)

y = tahmin

if x < maxiter:

y = (1/2)\*(tahmin + (sayi/tahmin))

if abs((y\*\*2) - (sayi)) < tol:

return "Hata payı yeterince küçüldü. Sayınızın karekökü olarak hesaplandı:", y

else:

return karekok(sayi, y, maxiter, x+1)

else:

return "Yeterli denemede sayıya ulaşılmadı. Karekök değeri olarak hesaplandı:", y

def en\_buyuk\_ob(x:int,y:int):

if y==0:

return x

else:

return en\_buyuk\_ob(y,x%y)

def asal\_sayi(x:int,y=2):

if x ==y:

return True

elif x%y==0:

return False

return asal\_sayi(x,y+1)

def fibo\_dizisi(n, k=1, fibk=1, fibk1=0):

print(n, k, fibk, fibk1)

if k == n:

return

else:

fibo\_dizisi(n, k+1, fibk+fibk1, fibk)

print("┌──────────────────────────────────────────────────────┐")

print("│ Menü │")

print("├──────────────────────────────────────────────────────┤")

print("│ 1-K’nıncı En Küçük Elemanı Bulma │")

print("│ 2-En Yakın Çifti Bulma │")

print("│ 3-Bir Listenin Tekrar Eden Elemanlarını Bulma │")

print("│ 4-Matris Çarpımı │")

print("│ 5-Bir Text Dosyasındaki Kelimelerin Frekansını Bulma │")

print("│ 6-Liste İçinde En Küçük Değeri Bulma │")

print("│ 7-Karekök Fonksiyonu │")

print("│ 8-En Büyük Ortak Bölen │")

print("│ 9-Asallık Testi │")

print("│ 10-Daha Hızlı Fibonacci Hesabı │")

print("│ 11-Çıkış │")

print("└──────────────────────────────────────────────────────┘")

x=["1","2","3","4","5","6","7","8","9","10","11"]

y=True

root=tk.Tk()

root.title("Console Menu")

while y:

secim=input("Bir seçim giriniz:")

if secim=="1":

secim1=int(input("Bir tam sayı girin:"))

secim2=input("Bir liste girin:")

string\_cevir1=""

for i in secim2:

string\_cevir1=string\_cevir1+i

string\_cevir1=string\_cevir1[1:-1]

print(string\_cevir1)

listeye\_geri\_cevir1=string\_cevir1.split(",")

print(listeye\_geri\_cevir1)

listeyi\_inte\_cevir1=[]

for i in listeye\_geri\_cevir1:

listeyi\_inte\_cevir1.append(int(i))

print("Bu listenin",secim1,". küçük elemanı:",k\_kucuk(secim1,listeyi\_inte\_cevir1))

elif secim=="2":

secim3=int(input("Lütfen bir tam sayı girin:"))

secim4=input("Bir liste girin")

string\_cevir2=""

for i in secim4:

string\_cevir2=string\_cevir2+i

string\_cevir2=string\_cevir2[1:-1]

listeye\_geri\_cevir2=string\_cevir2.split(",")

listeyi\_inte\_cevir=[]

for i in listeye\_geri\_cevir2:

listeyi\_inte\_cevir.append(int(i))

print(en\_yakin\_cift(secim3,listeyi\_inte\_cevir))

elif secim=="3":

secim5=input("Lütfen bir liste girin:")

string\_cevir=""

for i in secim5:

string\_cevir=string\_cevir+i

string\_cevir=string\_cevir[1:-1]

listeye\_geri\_cevir=string\_cevir.split(",")

print(tekrar\_eden\_elemanlar(listeye\_geri\_cevir))

elif secim =="4":

secim10=input("Lütfen bir liste girin")

secim11=input("Lütfen ikinci listeyi giriniz:")

string\_cevir4=""

for i in secim10:

string\_cevir4=string\_cevir4+i

string\_cevir4=string\_cevir4[1:-1]

listeye\_geri\_cevir4=string\_cevir4.split(",")

string\_cevir5=""

for i in secim11:

string\_cevir5=string\_cevir5+i

string\_cevir5=string\_cevir5[1:-1]

listeye\_geri\_cevir5=string\_cevir4.split(",")

print("Sonuç:",matris(listeye\_geri\_cevir4,listeye\_geri\_cevir5))

elif secim =="5":

secim12=input("Lütfen dosyanın adını uzantısını belirterek girin:")

print("Kelime frekansları:",kelime\_sayisi(secim12))

elif secim == "6":

secim6=input("Lütfen bir liste girin:")

string\_cevir3=""

for i in secim6:

string\_cevir3=string\_cevir3+i

string\_cevir3=string\_cevir3[1:-1]

listeye\_geri\_cevir3=string\_cevir3.split(",")

print("Bu listedeki en küçük değer:",en\_kucuk\_deger(listeye\_geri\_cevir3))

elif secim =="7":

secim13=int(input("Lütfen bir sayı girin:"))

secim14= float(input("Lütfen bir tahmin girin:"))

secim15=int(input("Lütfen bir deneme sayısı girin:"))

print("Sonuç:",karekok(secim13,secim14,secim15))

elif secim=="8":

secim7=int(input("Lütfen ilk sayıyı girin:"))

secim8=int(input("Lütfen ikinci sayıyı girin:"))

print("Bu iki sayının en büyük ortak böleni:",en\_buyuk\_ob(secim7,secim8))

elif secim=="9":

secim9=int(input("Lütfen bir tam sayı girin:"))

if asal\_sayi(secim9)==True:

print(secim9,"bir asal sayıdır.")

elif asal\_sayi(secim9)==False:

print(secim9,"bir asal sayı değildir.")

elif secim =="10":

secim16=int(input("Lütfen sayı girin:"))

print("Sonuç",fibo\_dizisi(secim16))

elif secim=="11":

print("İyi günler dileriz")

break

else:

print("Lütfen seçeneklerdeki değerlerden birini girin!")

continue

# 6. KAYNAKÇA

https://stackoverflow.com

https://medium.com/

https://blogs.sas.com

https://www.sanfoundry.com

https://chat.openai.com