به نام خداوند بخشنده و مهربان

پروژه اول دوره یادگیری ماشین و هوش مصنوعی در مهندسی – آزمایشگاه گرافن و مواد پیشرفته (گام لب)

مهلت ارسال پروژه: ساعت پنج بعد از ظهر یازده شهریور ماه

مهلت پاسخگویی به سوالات در مورد پروژه : تا دوازده شب جمعه نه شهریور ماه

توضیحات :
این پروژه سه قسمت دارد

قسمت اول

بر اساس رشته تحصیلی و زمینه مورد علاقه تان پنج عدد ثابت تعریف کنید و در پنج متغیر (ظرف) ذخیره سازید

قسمت دوم

بر اساس رشته تحصیلی و زمینه مورد علاقه تان دو تابع با توضیحات و فرمتی که در ادامه نوشته شده است بنویسید.

قسمت سوم

یک تابع تبدیل کننده بنویسید (دو تابع دو طرفه باشیند یعنی یک تابع مثل کیلومتر به متر . و تابع بعدی متر به کیلومتر)

نکات:

قبل از نوشتن تابع حتما جدول توابع در تلگرام را مشاهده فرمایید که تابع تکراری انتخاب نکنید.

اسم تابع ها را بر اساس کارکردی که دارند بنویسید و اگر یک کلمه است فقط حرف اول آن بزرگ اگر چند کلمه هست با استفاده از آندرلاین جدا کرده و حروف هر کلمه را بزرگ بنویسید
مثال Estimation
Newton_Law

درون تابع همانطور گفته شده است با استفاده از سه تا کوتیشن و اینتر توضیحات به زبان انگلیسی اضافه کنید که چه نوع ورودی و چه نوع خروجی ای میدهد

تابع ها باید فقط return داشته باشد
به این معنی که نباید از پرینت یا بدون خروجی ستفاده کنید که زمانی که آنها را صدا زدیم به ما محاسبات انجام شده در بدنه تابع را به صورت خروجی برگرداند.

فایل ارسال شده باید فقط حاوی اعداد و توابع باشد و هیچ گونه خط اضافی اعم از صدا زدن تابع و .. نداشته باشد و بیرون از تابع هیچ چیزی نوشته نشده باشد و فقط آماده ی صدا زدن باشند و همچنین از اینپوت و .. در بیرون یا داخل تابع استفاده ننمایید.

اخطار

به هیچ عنوان در ساختن کد یا نوشتن فرمول از چت بات های مبتنی بر هوش مصنوعی (اعم از چت جی پی تی و کوپایلت و .... استفاده نفرمایید)

تمام کدها داخل چک بات خود سایت openain چک میشود و هرگونه aiflag مساوی است با تصحیح نشدن پروژه های بعدی

فقط برای گرفتن ایده میتوانید از آنها کمک بگیرید و یا به ایمیل بنده پیام بدهید.

طریقه ارسال

تمام سه قسمت را در یک فایل با فرمت گفته شده با سابجکت گفته شده به ایمیل همیشگی بنده ارسال فرمایید

فرمت :

A1_fname_lname

اسم و فامیلی مثال A1_Sina_Ahmadi

ایمیل : ai.2024.pilehvar@gmail.com

یک مثال از فایل تهیه شده :

مثال از یک فایل:

PART1

```
#Part1 ------------------------------------------------------------------------
#IE_Constants

k=1.96              #k=Z(0.05) Normal Standard Distribution
pf=0.6209           #f(P/F , i% , n) = 0.6209 for i=10% and n=5
ag=3.1936           #f(A/G , i% , n) = 3.1936 for i=18% and n=10
t_stu=0.9277        #T-Student Distribution for x=1.5 and n=30 (degree freedom 29)
d=1.128             #Constant coefficients for n=2
```

pART2

```
def Lattice_Parameter(r,structure):
    '''
    Parameters
    ----------
    r : float
        r is atomic radius of material. It is in terms of angestrom.
    structure : str
        Structure of material is face center cubic or body center cubic.

    Returns
    -------
    a : float
        a is lattice parameter of material in unit cell. It is in terms of angestrom.

    '''
    if structure=='fcc':
        a=4*r/(2**0.5)
    if structure=='bcc':
        a=4*r/(3**0.5)
    return a
```

```python
def PengRobinson(T = None,P = None,Tc = None,Pc = None,w = None,MW = None,Phases = None):
    """
    PengRobinson.m : calculates the compressibility factor,fugacity coefficient and density
    of a pure compound with the Peng Robinson equation of state (PR EOS)

    Parameters
    ----------
    T : float
        Temperature [=] K
    P : float
        Presure [=] Pa
    Tc : float
        Critical temperature [=] K
    Pc : float
        Critical presure [=] Pa
    w : float
        Accentic factor
    MW : float
        Molar weigth [=] kg/mol.
    Phases : int
        if Phases == 1, then calculates liquid fugacity;
        if Phases == 0 then calculates vapor fugacity

    Returns
    -------
    Z : flout
        Compressibility factor
    fhi : float
        Fugacity coefficient
    density : float
        Density

    """
    R = 8.314

    # Reduced variables
    Tr = T / Tc

    # Parameters of the EOS for a pure component
    m = 0.37464 + 1.54226 * w - 0.26992 * w ** 2
    alfa = (1 + m * (1 - np.sqrt(Tr))) ** 2
    a = 0.45724 * (R * Tc) ** 2 / Pc * alfa
    b = 0.0778 * R * Tc / Pc
    A = a * P / (R * T) ** 2
    B = b * P / (R * T)
    # Compressibility factor
    Z = np.roots(np.array([1,- (1 - B),(A - 3 * B ** 2 - 2 * B),- (A * B - B ** 2 - B ** 3)]))
    ZR = []
    for i in range(3):
        if type(Z[i])!='complex':
            ZR.append(Z[i])

    if Phases == 1:
        Z = np.amin(ZR)
    else:
        Z = np.amax(ZR)

    # Fugacity coefficient
    fhi = np.exp(Z - 1 - np.log(Z - B) - A / (2 * B * np.sqrt(2)) * np.log((Z + (1 + np.sqrt(2)) * B) / (Z + (1 - np.sqrt(2)) * B)))
    if True:
        density = P * MW / (Z * R * T)
        result = np.array([Z,fhi,density])
    else:
        'No real solution for "fhi" is available in this phase'
        result = np.array(['N/A','N/A','N/A'])


    return Z,fhi,density
```

PART3 : Convertor

```python
def Liter_To_Cubic_Meter(number_in_Liter):
    '''
    This function converts liters to cubic meters.

    Parameters
    ----------
    number_in_Liter : int or float
        Number per liter unit.
    Cubic_Meter : int or float
        Number per cubic meter unit.

    '''
    Cubic_Meter= number_in_Liter/1000
    return (Cubic_Meter)
```