```python
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 14 16:32:05 2019

@author: VACALDER
"""

# -----------------------------------------------------------------------------
# |      PROGRAM TO CHECK TIME DEPENDENT PROPERTIES EFFECTS ON STRUCTURES      |
# |
# |          version: 2.0.2
# |          Victor A Calderon
# |          PhD Student/ Research Assistant
# |          NC STATE UNIVERSITY
# |          2021 (c)
# |
# |
# -----------------------------------------------------------------------------


# -----------------------------------------------------------------------------
# |                               IMPORTS
# -----------------------------------------------------------------------------

# import the os module
import pandas as pd
import time

start_time = time.time()
import os
from LibUnitsMUS import *
import Build_RC_Column
import Postprocessor_of_data
import openseespy.opensees as ops

# -----------------------------------------------------------------------------
# | VARIABLES THAT CHANGE WITH TIME
# -----------------------------------------------------------------------------
#
#
# *cover = Cover of concrete in cm
# *Tcorr = Time to corrosion in yrs
# *Time  = Different times that are being analyzed
# *wcr   = Water to cement ratio
# *dbi   = Initial longitudinal bar diameter
# *dti   = Initial transverse steel diameter


compressive_strength_concrete = 5 * ksi
yield_strength_long_steel = 60 * ksi
yield_strength_trans_steel = 60 * ksi
iShapeFactor = [4,6,8]
iCL = [0,5,10,15,20]
iCLt = [0]
pid = ops.getPID()
np = ops.getNP()
```

```python
GM_Path = r'/home/vacalderon/Documents/MainshocksParallel_2.0.3/GroundMotion_Mainshock_Records'
GMListing = os.listdir(GM_Path)
rootdir = r'/home/vacalderon/Documents/MainshocksParallel_2.0.3'
iALR = [0.05, 0.10, 0.15, 0.20]  # [0.10] #
GMDB = pd.read_csv('mainshock_file_database.csv')
GeomDB = pd.read_csv('column_database.csv')
counter = 0
# -----------------------------------------------------------------------
# |                              BATCH RUN
# -----------------------------------------------------------------------

for column, Crow in GeomDB.iterrows():
    D = float(Crow['column_diameter'])
    dbi = float(Crow['long_bar_diameter'])
    nbi = float(Crow['number_of_bars_longitudinal'])
    dti = float(Crow['trans_bar_diameter'])
    sti = float(Crow['spacing_trans_steel'])
    rhol = float(Crow['rho_l'])
    rhov = float(Crow['rho_v'])
    for shapefactor in iShapeFactor:
        Height_of_Column = shapefactor * D
        for ALR in iALR:

            Ag = 0.25 * math.pi * D ** 2
            AxialLoad = compressive_strength_concrete * Ag * ALR

            for GM, row in GMDB.iterrows():
                i = -1
                GM_fn = row['horizontal_1_filename']
                GM_dt = row['dt_horizontal1']
                GM_npt = row['npt_horizontal1']
                print('GM = ', GM_fn)
                GM_file = GM_Path + '/' + GM_fn
                for CL in iCL:
                    for CLt in iCLt:
                        if (counter % np) == pid:

                            datadir = rootdir + "/" + "data" + "/" + GM_fn + "/CL" + str(CL) + "/CL
                                shapefactor) + "/ALR" + str(ALR) + "/RhoL" + str(rhol) + "/Rhov" +
                            if not os.path.exists(datadir):
                                os.makedirs(datadir)

                            if yield_strength_long_steel == 60*ksi:
                                alpha = 0.0075
                            elif yield_strength_long_steel == 80*ksi:
                                alpha = 0.0075

                            dblc = ((1 - CL*0.01) ** 0.5) * dbi
                            Build_RC_Column.Build_RC_Column(D, Height_of_Column, compressive_streng
                                                            yield_strength_long_steel, yield_streng
                                                            dbi, dti, CL, dblc, nbi, CLt, sti, data
                                                            GM_file, GM_dt, GM_npt, ALR,alpha)

                            with open(datadir + "/Conditions.out", 'w') as f:
                                f.write("%s \n" % (CL))
                            f.close
```

```python
                        Postprocessor_of_data.Postprocessor_of_data(GM_fn, CL, CLt, D, shapefac

                        os.remove(datadir + "/StressStrain.out")
                        os.remove(datadir + "/StressStrain2.out")
                        os.remove(datadir + "/StressStrain3.out")
                        os.remove(datadir + "/StressStrain4.out")
                        os.remove(datadir + "/Conditions.out")
                        os.remove(datadir + "/DFree.out")
                        os.remove(datadir + "/mat.out")
                        os.remove(datadir + "/Period.out")
                        os.remove(datadir + "/PGA.out")
                        os.remove(datadir + "/RBase.out")
                counter += 1
print("ALL ANALYSIS COMPLETE")
print("--- %s minutes ---" % ((time.time() - start_time) / 60))
```