```python
__author__ = 'Geoffrey Nyaga'

import sys
sys.path.append('../')
from API.db_API import write_to_db, read_from_db

import numpy as np
import matplotlib.pylab as plt

a = np.arange(50)

ws = np.arange(10, 35, .01)

cdmin = 0.025
write_to_db('cdMin',cdmin)

do = read_from_db('rhoSL')
dalt = read_from_db('altitudeDensity')   #AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
k = read_from_db('k')

# v = read_from_db('cruiseSpeed') * 1.688
v = (read_from_db("maxSpeed") * 1.688)/1.2 #AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
print("v",v)
qcruise = 0.5 * dalt * v ** 2  # dynamic pressure at cruise
qtakeoff = 0.5 * do * v ** 2  # dynamic pressure at take-off

turnangle = 40  # turn angle
loadfactor = 1 / (np.cos(turnangle))  # loadfactor
twturn = qcruise * ((cdmin / ws) + (k * (loadfactor / qcruise) ** 2) * ws) * (v * 5850 / (0.8 * 550 * 0.6604))

# rate of climb
roc = read_from_db('rateOfClimb') * 3.28 * 60 # rate of climb ft/min   # AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
# Vy=sqrt((2/do)*ws * sqrt( k/(3*cdmin) ))
Vy = 150
Vv = roc / 60
qclimb = 0.5 * do * (Vy ** 2)
twclimb = ((Vv / Vy) + ((qclimb / ws) * cdmin) + ((qclimb / ws) * cdmin) + ((k / qclimb) * ws)) * (
Vy * 5850 / (0.6 * 550))

# ground run
Sg = 900  # ground run ft
Vlof = 70 * 1.688
clto = 1.08
u = 0.04
cdto = 0.03
q1 = 0.5 * do * (Vlof / np.sqrt(2)) ** 2
twtakeoff = (((Vlof ** 2) / (2 * 32.174 * Sg)) + ((q1 * cdto) / ws) + u * (1 - (q1 * clto / ws))) * (
Vlof * 5850 / (0.6 * 550))

# cruise altitude
twcruise = (((qcruise * cdmin) / ws) + ((k / qcruise) * ws)) * (v * 5850 / (0.6 * 550 * 0.6604))

# service ceiling
twservceiling = ((1.668 / np.sqrt((2 * ws / dalt) * np.sqrt(k / (3 * cdmin)))) + (4 * np.sqrt(k * cdmin / 3))) * (
(v * 5850) / (0.7 * 550 * 0.6604))

plt.plot(ws, twclimb, label = 'climb')
plt.plot(ws, twturn,label = 'turn')
plt.plot(ws, twtakeoff,label = 'Takeoff')
plt.plot(ws, twservceiling,label = 'Service Ceiling')
plt.plot(ws, twcruise,label = 'cruise')
plotWS = read_from_db('WS')
plt.axvline(x=plotWS)                         ################################
plt.legend(loc='upper left')

if __name__ == '__main__':
    plt.show()



def find_nearest(array,value):
    idx = (np.abs(array-value)).argmin()
    return idx

```

```
# print(find_nearest(ws, plotWS))
myidx = find_nearest(ws, plotWS)

# cruiseidx = (twcruise[myidx])
# takeoffidx = twtakeoff[myidx]
# climbidx = twclimb[myidx]
# turnidx = twturn[myidx]
# ceilingidx = twservceiling[myidx]
# print([cruiseidx,takeoffidx,climbidx,turnidx,ceilingidx])

def point():
    cruiseidx = (twcruise[myidx])
    takeoffidx = twtakeoff[myidx]
    climbidx = twclimb[myidx]
    turnidx = twturn[myidx]
    ceilingidx = twservceiling[myidx]
    # print([cruiseidx,takeoffidx,climbidx,turnidx,ceilingidx])
    # print (cruiseidx,"cruiseidx")

    x = np.array([cruiseidx,takeoffidx,climbidx,turnidx,ceilingidx])
    idx = x.argmax()
    return x[idx]

finalBHP= point()
# print ( finalBHP,"BHP")

write_to_db('finalBHP',finalBHP)

S=(read_from_db('finalMTOW'))/(plotWS*10.57)
write_to_db('S',S)
```