

```
#Author: 7_hermit
#Mail: uzumaki.sahoo@gmail.com
import math as m

class No_load:
    def __init__(self, v, i, p, s):
        self.__input_voltage = v
        self.__input_current = i
        self.__input_power = p
        self.__rpm = s
        self.__power_factor = 0.0
        self.__iw = 0.0
        self.__im = 0.0
        self.__resistance = 0.0
        self.__reactance = 0.0

    def calculate(self):
        self.__power_factor = (self.__input_power) / (self.__input_voltage * self.__input_current)
        self.__iw = self.__input_current * self.__power_factor
        self.__im = self.__input_current * m.sqrt(1 - self.__power_factor * self.__power_factor)
        self.__resistance = self.__input_voltage/self.__iw
        self.__reactance = self.__input_voltage/self.__im

    def get_magnetizing_current(self):
        return self.__im

    def get_no_load_resistance(self):
        return self.__resistance

    def get_no_load_reactance(self):
        return self.__reactance

    def get_slip(self):
        return ((1500-self.__rpm)/1500)

class Blocked_rotor:
    def __init__(self, v, i, p, r):
        self.__input_voltage = v
        self.__input_current = i
        self.__input_power = p
        self.__motor_equivalent_impedance_referred_stator = 0.0
        self.__motor_equivalent_resistance_referred_stator = 0.0
        self.__power_factor = 0.0
        self.__motor_equivalent_reactance_referred_stator = 0.0
        self.__stator_resistance = r
        self.__stator_reactance = 0.0
        self.__rotor_resistance_referred_stator = 0.0
        self.__rotor_reactance_referred_stator = 0.0
        self.__magnetizing_reactance = 0.0

    def calculate(self, no_load_reactance):
```

```

        self.__motor_equivalent_impedance_referred_stator = self.__input_voltage / self.__input_current
        self.__power_factor = (self.__input_power) / (self.__input_voltage * self.__input_current)
        self.__motor_equivalent_resistance_referred_stator = self.__motor_equivalent_impedance_referred_stator * self.__power_factor
        self.__motor_equivalent_reactance_referred_stator = self.__motor_equivalent_impedance_referred_stator * m.sqrt(1 - m.pow(self.__power_factor, 2))
        self.__rotor_resistance_referred_stator = self.__motor_equivalent_resistance_referred_stator - self.__stator_resistance
        self.__rotor_reactance_referred_stator = self.__stator_reactance = self.__motor_equivalent_reactance_referred_stator * 0.5
        self.__magnetizing_reactance = 2*(no_load_reactance - self.__stator_reactance - 0.5 * self.__rotor_reactance_referred_stator)

    def get_magnetizing_reactance(self):
        return self.__magnetizing_reactance

    def get_rotor_resistance_referred_stator(self):
        return self.__rotor_resistance_referred_stator

    def get_rotor_reactance_referred_stator(self):
        return self.__rotor_reactance_referred_stator

    def get_stator_resistance(self):
        return self.__stator_resistance

    def get_stator_reactance(self):
        return self.__stator_reactance

def main(nl, br):
    no_load = No_load(nl[0], nl[1], nl[2], nl[3])
    no_load.calculate()
    R0 = no_load.get_no_load_resistance()
    X0 = no_load.get_no_load_reactance()
    slip = no_load.get_slip()

    blocked_rotor = Blocked_rotor(br[0], br[1], br[2], br[3])
    blocked_rotor.calculate(X0)

    Xm = blocked_rotor.get_magnetizing_reactance()
    R1 = blocked_rotor.get_stator_resistance()
    X1 = blocked_rotor.get_stator_reactance()
    R21 = blocked_rotor.get_rotor_resistance_referred_stator()
    X21 = blocked_rotor.get_rotor_reactance_referred_stator()

    elements = [R0, X0, R1, X1, R21, X21, Xm, slip]
    return elements

if __name__ == '__main__':

    no_load_data = [220, 5.9, 260, 1492] #[v, A, W, rpm]
    blocked_rotor_data = [60, 7.5, 400, 4.08] #[v, A, w, ohms]

    elements = main(no_load_data, blocked_rotor_data)
    print('No Load Registance(ohms):', elements[0])
    print('No Load Reactance(ohms):', elements[1])
    print('Stator Registance(ohms):', elements[2])

```

```
print('Stator Reactance (ohms):', elements[3])
print('Rotor Registance referred to stator (ohms):', elements[4])
print('Rotor Reactance referred to stator (ohms):', elements[5])
print('Magnetizing Reactance (ohms):', elements[6])
print('Slip:', elements[7])
```