#Imports  
import pandas as pd  
from collections import defaultdict, OrderedDict

#Data Reading and Formatting  
directory = r'(filepath to data)'  
OD\_readings = pd.read\_excel('%s\input\_data.xlsx'%directory, sheet\_name = 'OD\_readings')  
MIC\_metadata = pd.read\_excel('%s\input\_data.xlsx'%directory, sheet\_name = 'MIC\_metadata')  
OD\_readings = OD\_readings.reset\_index(drop=True).set\_index('Drugs')  
MIC\_metadata = MIC\_metadata.reset\_index(drop=True).set\_index('Drugs').T

#Grouping the Replicates  
grouped\_metadata = defaultdict(dict)  
for antibiotic in MIC\_metadata.columns[:11]:  
    for concentration in set(MIC\_metadata[antibiotic]):  
        grouped\_metadata[antibiotic][concentration] = MIC\_metadata.loc[MIC\_metadata[antibiotic] == concentration].index.tolist()

#Grouping the Readings  
grouped\_readings = defaultdict(dict)  
for antibiotic in OD\_readings.index:  
    for concentration in grouped\_metadata[antibiotic].keys():  
        columns = grouped\_metadata[antibiotic][concentration]   
        grouped\_readings[antibiotic][concentration] = OD\_readings.loc[antibiotic, columns].values  
grouped\_readings = {antibiotic: OrderedDict(sorted(d.items(), key = lambda a: a[0])) for antibiotic, d in grouped\_readings.items()}

#Function to treat special cases  
def treat\_special(reading\_array, cutoff = 0.15, minimum = 0.135, maximum = 0.165):  
    if(sum(reading\_array>cutoff)==2):  
        if(reading\_array[reading\_array <= cutoff][0]>=minimum):  
            if(i+1 < len(readings)):  
                MIC = '%s'%str(list(readings.keys())[i+1])  
            else:  
                MIC = '>%s'%str(key)  
        else:  
            if(i+1 < len(readings)):  
                MIC = '%s-%s'%(str(list(readings.keys())[i]), float(list(readings.keys())[i+1]))  
            else:  
                MIC = '%s'%str(list(readings.keys())[i])  
    elif(sum(reading\_array>cutoff)==1):  
        if(reading\_array[reading\_array > cutoff][0]<maximum):  
            MIC = '%s'%str(list(readings.keys())[i])  
        else:  
            if(i+1 < len(readings)):  
                MIC = '%s-%s'%(str(list(readings.keys())[i]), float(list(readings.keys())[i+1]))  
            else:  
                MIC = '>%s'%str(list(readings.keys())[i])  
    return(MIC)

#Finding the MICs  
cutoff = 0.15  
minimum = 0.135  
maximum = 0.165  
resulting\_MICs = {}  
for antibiotic, readings in grouped\_readings.items():  
    res = list(readings.keys())  
    for i, key in enumerate(readings.keys()):  
        reading\_concentration = list(readings.keys())[i]  
        reading\_array = readings[key]  
        #print(reading\_concentration ,(reading\_array>cutoff))  
        if(reading\_array>cutoff).all():  
            res.remove(reading\_concentration)  
            #print('All true')  
        elif not(reading\_array<=cutoff).all():  
            #print('special')  
            resulting\_MICs[antibiotic] = treat\_special(reading\_array)  
            break  
        #print(antibiotic,res)  
  
        if len(res) == 0:  
             resulting\_MICs[antibiotic] = '>%s'%str(max(readings.keys()))  
  
        else:        
            resulting\_MICs[antibiotic] = '%s'%str(min(res))

#Output to CSV  
human\_readable = pd.DataFrame(list(resulting\_MICs.items()), columns = ['Antibiotic', 'MIC'])  
human\_readable = human\_readable.set\_index('Antibiotic').loc[MIC\_metadata.columns.tolist()].reset\_index().drop\_duplicates()  
human\_readable.to\_csv('%s\MICs for (Insert Name).csv'%directory, index=None)