

ID: A1809

Date of Submission : 30-03-19

# Entrance Challenge: When Will the Sakura Bloom?

Importing Libraries

In [27]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
import datetime as dt
from sklearn.metrics import mean_squared_error
from sklearn.neural_network import MLPRegressor
from sklearn.preprocessing import normalize
from sklearn import preprocessing
import warnings
from copy import deepcopy
warnings.filterwarnings('ignore')
```

## 0. Basics of the Sakura Bloom-cycle (5pts total)

In a year, sakura trees basically go through 4 phases: energy production, hibernation, growth, and of course flowering. These phases roughly follow the seasons, but not exactly.

Production phase : Initial development of the buds (Summer-Fall)

Hibernation phase : Bud growth stops while the tree goes into hibernation (Late Fall-Winter)

Growth phase : Buds once again continue to grow when the tree comes out of its winter hibernation (Late Winter-Spring)

Flowering phase : The buds finally bloom in spring (as climate conditions allow), once they have been able to fully develop.  
(Spring)

Each year, near the end of winter but before the trees finally bloom, the hibernation period ends. The sakura that rested through the winter once gain become metabolically active, and the buds continue to grow (though we may not immediately notice when this happens.) However, the cycle is not simply clockwork- for example, in places where the temperature is above 20°C year-round, the trees are unable to hibernate sufficiently, and thus cannot blossom.

In this challenge, we have outlined the basic mechanism by which the sakura reach their eventual bloom-date. We consider building a bloom-date prediction model for the case of sakura in Tokyo, with the data split as follows:

Test years : 1966, 1971, 1985, 1994, and 2008

Training years: 1961 to 2017 (Excluding the test years)

You should fit the model to the data from the training years, then use the model to predict the bloom-date for each of the test years. The 3 models to be applied to the data are described below.

### Problem 0-1: (5pts)

Acquire data of sakura blooming date (桜の開花日) for Tokyo from 1961 to 2018 using the Japanese Meteorological Agency website (気象庁).

In [2]:

```
sakura = pd.read_csv("tokyo.csv")
BloomDate = pd.read_csv("BloomDate.csv")
sakura.rename(columns = lambda x: x.replace(' ', '_').lower(), inplace=True)
BloomDate.rename(columns = lambda x: x.replace(' ', '_').lower(), inplace=True)
```

## 1. Prediction using the "600 Degree Rule" (15pts total)

## 1. Prediction using the 600 Degree Rule (15pts total)

For a rough approximation of the bloom-date, we start with a simple "rule-based" prediction model, called the "600 Degree Rule". The rule consists of logging the maximum temperature of each day, starting on February 1st, and sum these temperatures until the sum surpasses 600°C. The day that this happens is the predicted bloom-date. This 600°C threshold is used to easily predict bloom-date in various locations varies by location. However, for more precise predictions, it should be set differently for every location. In this challenge, we verify the accuracy of the "600 Degree Rule" in the case of Tokyo.

### Problem 1-1: (5pts)

From here-on, we refer to the bloom-date in a given year  $y$  as  $BD_y$ . For each year in the training data, calculate the accumulated daily maximum temperature from February 1st to the actual bloom-date  $BD_y$ , and plot this accumulated value over the training period. Then, average this accumulated value as  $T_{mean}$ , and verify whether we should use 600°C as a rule for Tokyo.

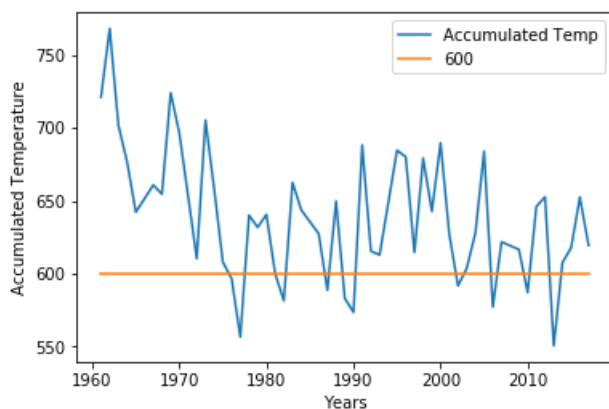
In [63]:

```
train = pd.read_csv("train.csv")
year_wise_accumulated_temp = []
years = train["year"].unique()
for year in years:
    add = 0
    df = train.loc[(train["year"]==year) & (train["month"]!=1)]

    for row,bl in df.iterrows():
        if bl['year'] not in [1966, 1971, 1985, 1994, 2008]:
            temp_add = (df.loc[row,"highest_temp"])# + df.loc[row,"lowest_temp"])/2
            #if temp_add>600:
            #    temp_add-=600
            #    print("hello")
            add+=temp_add
            if bl["bloom"] == 1:
                break
    year_wise_accumulated_temp.append(add)
    #print(year,add)

Tmean = np.mean(year_wise_accumulated_temp)
print("Tmean = ",Tmean)
plt.plot(years,year_wise_accumulated_temp,label = "Accumulated Temp")
plt.plot(years,[600 for i in range(len(years))],label = "600")
#plt.plot(years,[600 for i in range(len(years))], "b",label="600 Degree")
plt.legend()
plt.xlabel("Years")
plt.ylabel("Accumulated Temperature")
plt.show()
# As the Tmean value is around 600 and from the plot it is clear that maximum year wise
# accumulated temperature
# is around 600 so it can be used as a rule for Tokyo.
```

Tmean = 638.3557692307693



### Problem 1-2: (10pts)

Use the average accumulated value  $\overline{BD_{ij}}$  calculated in 1-1 to predict  $BD_{ij}$  for each test year, and show the error from the actual  $BD_{ij}$ . Compare to the prediction results when 600 is used as a threshold value, and evaluate both models using the coefficient of determination ( $R^2$  score).

In [66]:

```
rule_based_threshold = 600
test = pd.read_csv("test.csv")
actual = test[test["bloom"]==1]
years = actual["year"]
predicted = []
predicted_600 = []
#print(years)

#calculating leap years

def is_leap_year(year):
    """ if year is a leap year return True
        else return False """
    if year % 100 == 0:
        return year % 400 == 0
    return year % 4 == 0

#converting date of year

def doy(Y,M,D):
    """ given year, month, day return day of year
        Astronomical Algorithms, Jean Meeus, 2d ed, 1998, chap 7 """
    if is_leap_year(Y):
        K = 1
    else:
        K = 2
    N = int((275 * M) / 9.0) - K * int((M + 9) / 12.0) + D - 30
    return N

for year in years:
    add=0
    df = test.loc[(test["year"]==year) & (test["month"]!=1)]
    for row,bl in df.iterrows():
        add+=df.loc[row,"highest_temp"]
        if add>Tmean:
            conv_day = doy(bl["year"],bl["month"],bl["day"])
            print("Blooming Day", conv_day)
            predicted.append(conv_day)
            break

#for 600
for year in years:
    add=0
    df = test.loc[(test["year"]==year) & (test["month"]!=1)]
    for row,bl in df.iterrows():
        add+=df.loc[row,"highest_temp"]
        if add>rule_based_threshold:
            conv_day = doy(bl["year"],bl["month"],bl["day"])
            predicted_600.append(conv_day)
            break

actual_val = []
for row,bl in actual.iterrows():
    conv_day = doy(bl["year"],bl["month"],bl["day"])
    actual_val.append(conv_day)

actualValueForTmean=deepcopy(actual_val)
predictedValueForTmean=deepcopy(predicted)

#for 638
print("R2 score for 638 degree: ", r2_score(actual_val,predicted))
#for 600
print("R2 score for 600 degree: ", r2_score(actual_val,predicted_600))

print('Actual: ',actual_val)
print('Predicted for Tmean', predicted)
```

```
print('Predicted for 600', predicted_600)
```

```
Blooming Day 82.0
Blooming Day 88.0
Blooming Day 91.0
Blooming Day 90.0
Blooming Day 85.0
R2 score for 638 degree: 0.8323615160349854
R2 score for 600 degree: 0.6793002915451896
Actual: [79.0, 89.0, 93.0, 90.0, 82.0]
Predicted for Tmean [82.0, 88.0, 91.0, 90.0, 85.0]
Predicted for 600 [79.0, 86.0, 88.0, 87.0, 83.0]
```

The R squared value for Tmean is much better than just using 600 as threshold. So it seems Tmean works better for this particular test years. Now we should try it for other test years as well. After that we can give some opinions.

## 2. Linear Regression Model: Transform to Standard Temperature (30pts total)

The year to year fluctuation of the bloom-date depends heavily upon the actual temperature fluctuation (not just the accumulated maximum). In order to get to a more physiologically realistic metric, Sugihara et al. (1986) considered the actual effect of temperature on biochemical activity. They introduced a method of "standardizing" the temperatures measured, according to the fluctuation relative to a standard temperature.

In order to make such a standardization, we apply two major assumptions, outlined below.

### 1) The Arrhenius equation:

The first assumption, also known in thermodynamics as the "Arrhenius equation", deals with chemical reaction rates and can be written as follows:

$$k = A \exp\left(-\frac{E_a}{RT}\right)$$

Basically, it says that each reaction has an activation energy,  $E_a$  and a pre-exponential factor  $A$ . Knowing these values for the particular equation, we can find the rate constant  $k$  if we know the temperature,  $T$ , and applying the universal gas constant,  $R = 8.314 \text{ J/K} \cdot \text{mol}$ .

### 2) Constant output at constant temperature:

The second assumption, is simply that the output of a reaction is a simple product of the duration and the rate constant  $k$ , and that product is constant even at different temperatures.

$$tk = t'k' = t''k'' = \dots = \text{const}$$

Making the assumptions above, we can determine a "standard reaction time",  $t_s$  required for the bloom-date to occur. We can do so in the following way:

$$t_s = \exp\left(\frac{E_a}{R(T_{i,j} - T_s)}\right)$$

We define  $T_{i,j}$  as the daily average temperature, and use a standard temperature of  $T_s = 17^\circ\text{C}$ . For a given year  $j$ , with the last day of the hibernation phase set as  $D_j$ , we define the number of "transformed temperature days",  $DTS_j$ , needed to reach from  $D_j$  to the bloom-date  $BD_j$  with the following equation:

$$DTS_j = \sum_{i=D_j}^{BD_j} t_s = \sum_{i=D_j}^{BD_j} \exp\left(\frac{E_a}{R(T_{i,j} - T_s)}\right)$$

From that equation, we can find the average  $DTS$  for  $x$  number of years ( $DTS_{\text{mean}}$ ) as follows:

$$DTS_{\text{mean}} = \frac{1}{\sum_{j=1}^x DTS_j} \sum_{j=1}^x DTS_j = \frac{1}{\sum_{j=1}^x \sum_{i=D_j}^{BD_j} \exp\left(\frac{E_a}{R(T_{i,j} - T_s)}\right)}$$

In this exercise, we assume that  $DTS_{\text{mean}}$  and  $E_a$  are constant values, and we use the data from the training years to fit these 2 constants. The exercise consists of 4 steps:

1. Calculate the last day of the hibernation phase  $D_{\{j\}}$  for every year  $j$ .
2. For every year  $j$ , calculate  $DTS_{\{j\}}$  as a function of  $E_{\{a\}}$ , then calculate the average (over training years)  $DTS_{\{mean\}}$  also as a function of  $E_{\{a\}}$ .
3. For every year  $j$ , and for every value of  $E_{\{a\}}$ , accumulate  $t_{\{s\}}$  from  $D_{\{j\}}$  and predict the bloom date  $BD_{\{j\}}^{\{pred\}}$  as the day the accumulated value surpasses  $DTS_{\{mean\}}$ . Calculate the bloom date prediction error as a function of  $E_{\{a\}}$ , and find the optimal  $E_{\{a\}}$  value that minimizes that error.
4. Use the previously calculated values of  $D_{\{j\}}$ ,  $DTS_{\{mean\}}$ , and  $E_{\{a\}}$  to predict bloom-day on years from the test set.

## Problem 2-1: (5pts)

According to Hayashi et al. (2012), the day on which the sakura will awaken from their hibernation phase,  $D_{\{j\}}$ , for a given location, can be approximated by the following equation:

$$D_{\{j\}} = 136.75 - 7.689 \phi + 0.133 \phi^2 - 1.307 \ln L + 0.144 T_{\{F\}} + 0.285 T_{\{F\}}^2$$

where  $\phi$  is the latitude [ $^{\circ}$ N],  $L$  is the distance from the nearest coastline [km], and  $T_{\{F\}}$  is that location's average temperature [ $^{\circ}$ C] over the first 3 months of a given year. In the case of Tokyo,  $\phi = 35^{\circ}40'$  and  $L = 4$  km.

Find the  $D_{\{j\}}$  value for every year  $j$  from 1961 to 2017 (including the test years), and plot this value on a graph.

(In Problem 1, we had assumed a  $D_{\{j\}}$  of February 1st.)

In [53]:

```
#given from equation
latitude = 35 + 40/60

L = 4

all_data = pd.read_csv("tokyo.csv")

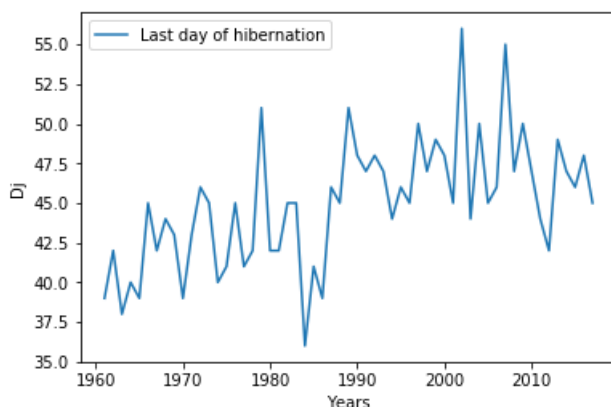
years = all_data["year"].unique()

Dj = []
all_yearly_Dj = []

for year in years:

    df = all_data.loc[(all_data["year"]==year) & (all_data["month"]<4)]
    #temperature
    Tf = np.mean(df["avg_temp"])
    #from equation
    dj = 136.75 - (7.689*latitude) + (0.133*(latitude**2)) - (1.307*np.log(L)) + (0.144*Tf) + (0.285
    *(Tf**2))
    #Dj.append(dj)
    Dj.append(np.round(dj))
    all_yearly_Dj.append((years, (np.round(Dj))))

#plotting the values
plt.plot(years,Dj,label = "Last day of hibernation")
plt.legend()
plt.xlabel("Years")
plt.ylabel("Dj")
plt.show()
print(Dj)
```



[39.0, 42.0, 38.0, 40.0, 39.0, 45.0, 42.0, 44.0, 43.0, 39.0, 43.0, 46.0, 45.0, 40.0, 41.0, 45.0, 41.0, 42.0, 51.0, 42.0, 42.0, 45.0, 45.0, 36.0, 41.0, 39.0, 46.0, 45.0, 51.0, 48.0, 47.0, 48.0, 47.0, 44.0, 46.0, 45.0, 50.0, 47.0, 49.0, 48.0, 45.0, 56.0, 44.0, 50.0, 45.0, 46.0, 55.0, 47.0, 50.0, 47.0, 44.0, 42.0, 49.0, 47.0, 46.0, 48.0, 45.0]

## Problem 2-2: (10pts)

Calculate  $DTS_j$  for each year  $j$  in the training set for discrete values of  $E_a$ , varying from 5 to 40kcal ( $E_a = 5, 6, 7, \dots, 40$  kcal), and plot this  $DTS_j$  against  $E_a$ . Also calculate the average of  $DTS_j$  over the training period, and indicate it on the plot as  $DTS_{mean}$ . Pay attention to the units of **every parameter** ( $T_{i,j}$ ,  $E_a$ , ...) in the equation for  $T_s$ .

In [67]:

```
train_data = pd.read_csv("train.csv")
years = train_data["year"].unique()
DTSj = []
DTSNewForTest=[]
dayCount=0
R = (8.314 * 10 ** -3)/4.184
Ts=17 + 273.15 #converting to kelvin
for Ea in range(5,41,1):

    for year in years:
        df = train_data.loc[(train_data["year"]==year)]
        df.reset_index(inplace=True)
        add=0
        dayCount=0
        row = Dj[year-1961]
        for index,r in df.iterrows():
            dayCount+=1
            if dayCount>=row:
                Tij = (273.15 + r['avg_temp'])
                add+= np.exp(((Tij-Ts)*Ea)/(R*Tij*Ts))
                if r['bloom']==1:
                    DTSj.append(add)
                    break

print(DTSj)
DTSmean=[]
for i in range(0,len(DTSj),52):
    c = DTSj[i:i+52]
    DTSmean.append(np.average(c))
#DTSmean.append(c/len(years))

for idx in range(0,37):
    g = [DTSj[idx] for idx in range(idx,len(DTSj),52)]
    plt.plot([x for x in range(5,41)],g,color="g")
    plt.xlabel("Energy, Ea")
    plt.ylabel("DTSj")

plt.plot([x for x in range(5,41)],DTSmean,label = "DTSmean",color="r")
plt.legend()
plt.show()
```

[38.910184020356446, 37.31596443870652, 39.57567169706983, 39.14795314917655, 38.855371519892174, 36.11973376156855, 34.73758126501, 40.369822603724316, 42.69855423700818, 32.47083194742553, 34.860776040521294, 38.896422279684735, 34.74045251977132, 29.363254220272953, 30.762719375903394, 36.15117510425791, 24.777359926237015, 37.02893416914124, 32.95157690169447, 29.41851341148468, 34.57189350047144, 46.815734097396664, 39.966767584366096, 27.617626831133055, 36.197550275950945, 22.58516028670156, 26.109897001428315, 33.31944824241389, 28.56923901843935, 28.071550455613522, 34.465473439213774, 35.712531495723745, 24.612551409424395, 30.926062178890543, 27.73561573690964, 32.61584510677082, 29.595479349578607, 16.77848712880796, 32.12973490354575, 23.52327690926109, 35.09726050176687, 27.304508706050868, 19.513846182709496, 24.292070264561524, 27.7434494765644, 33.15192599425089, 37.492503716379666, 21.21114574730047, 29.31467578085902, 28.558615193232747, 26.503728809845036, 27.51672283695065, 36.6194698047874, 35.22425568640038, 37.244009164114594, 36.76257434510222, 36.39734710579359, 34.19796482430731, 32.89491863858385, 38.162966699194776, 40.073649713737396, 30.738847731263473, 33.00805009121008, 36.61085579372689, 32.591589028153514, 27.910854852866336, 29.09832050499801, 34.05656286549209, 23.55722880243115, 34.905668130188495, 31.1425492071686, 27.982567151349595, 32.67043346852749, 43.63364791634816, 37.53545897968753, 26.07921352201157, 34.107930709400975, 21.5059132798516, 25.08188151717132, 31.70290065804905, 27.169990577605407, 26.5772354024305, 32.703902428738864, 33.855578094539304, 23.52977235662884, 29.399153612173805, 26.51540579874552, 30.895697416503705, 28.194952146826274, 16.211301365578485, 30.337846834688065, 22.58880261338847, 33.284621371084604, 26.00390563093442, 18.58797994342939,

23.167980834535296, 26.531504747926064, 31.354533000522178, 35.42928697983696, 20.256070486144537, 27.867771523176497, 27.139851276786555, 25.24821012790964, 26.090311304686253, 34.4768931331279, 3.25914453263344, 35.067157046291285, 34.54044936636022, 34.100249623311086, 32.40212115355593, 31.167183994027305, 36.10049741124022, 37.6257541459515, 29.112753519534962, 31.262818563189963, 34.47503240964717, 30.58403130940825, 26.53774464099482, 27.54055100310973, 32.09544151749682, 22.40211954353837, 32.915640721508225, 29.44630533419154, 26.627082385697257, 30.879217425921677, 40.686128270537836, 35.265038359613136, 24.636299467980525, 32.151017884336596, 20.48543291644216, 24.098480872088953, 30.177861028696668, 25.85250351651414, 25.166955397995263, 31.047178731927766, 32.11169413050688, 22.503245683163474, 27.955471710859598, 25.362158669503415, 29.27721493005891, 26.874475836750666, 15.667225269489547, 28.65489229841458, 21.70120076777414, 31.576849866957453, 24.772362993085945, 17.711855774102798, 22.106421157713413, 25.3878830552433, 29.66349614067755, 3.49065862808497, 19.35852532997234, 26.503927160247823, 25.798867242006533, 24.062742997351, 24.74218595060955, 32.47229331823198, 31.41253401056768, 33.03425282885286, 32.46994903550321, 31.953215526508775, 30.72299820937451, 29.546448866672183, 34.17199548337826, 35.34216686303934, 27.58561948880613, 29.618505551075998, 32.47846374037655, 28.70806318184521, 25.239275683581706, 2.608176911090967, 30.258670649358507, 21.308339745027887, 31.050055302035428, 27.855210188479063, 25.347194511349677, 29.191564396245273, 37.955005805639736, 33.14419551498451, 23.282532776810076, 30.317865286335337, 19.52027995358214, 23.157611939379215, 28.73871830395977, 24.61161119476998, 23.835775309710574, 29.473009845572495, 30.473359665354142, 21.52972902459704, 26.590211289812785, 24.271893637058092, 27.75398057955032, 25.629051230526684, 15.14525026930446, 27.07397707119827, 20.857912508694685, 29.967341407686153, 23.60593889052059, 16.88267672113553, 21.10365133098896, 24.308317714540344, 28.07219060621781, 31.66862855586332, 18.514787092565676, 25.21803209449353, 24.53112228058953, 22.943123925676833, 23.467879057595816, 30.596235886712353, 29.676867385564293, 31.135222672727874, 30.5403005032638, 29.94613450944468, 29.152123003141988, 28.02537136354387, 32.367842239953234, 33.211138614950805, 26.150993140245266, 28.06895197259338, 30.611432552614847, 26.954659017609117, 24.01108741708979, 24.7149237657084, 28.537770845689675, 20.272417987584635, 2.9.30071786665158, 26.36217655212992, 24.138348404044905, 27.60121388912511, 35.42356689594251, 31.16245299709573, 22.012003638384943, 28.60015562557473, 18.60722679984603, 22.25729405279943, 27.3802237533121, 23.44248894240468, 22.57905978312882, 27.99222912204858, 28.933591721189668, 20.606184251934938, 25.298856730887085, 23.24087755088961, 26.32000154698552, 24.454007361565253, 14.644415142864162, 25.588654435309163, 20.05652794501531, 28.449933461630945, 22.50092766797507, 16.097808934190965, 20.156166556535943, 23.28882543894227, 26.574418608650575, 29.9557433831093, 1.7.721385228049286, 24.0053037634757, 23.332353342359344, 21.885412175093148, 22.263185102286506, 2.8.839958990478337, 28.045090904231586, 29.36072270108378, 28.741522384284494, 28.06959606321618, 2.7.681692069058034, 26.5971505731885, 30.67915534126163, 31.221798029391586, 24.802864963920698, 26.608388431672722, 28.86493392386689, 25.31543387593924, 22.849088108762558, 23.433506555488872, 2.6.92487382413655, 19.291090009260092, 27.65999451778391, 24.96062397549136, 22.99627795202911, 26.102298544819526, 33.076433997423784, 29.310102503344503, 20.81921257321126, 26.99015510180253, 17.743243975431994, 21.395643779671, 26.097466733148323, 22.34063070773626, 21.39245454350078, 26.594172349475894, 27.485903761415308, 19.72976359516269, 24.07716393874826, 22.26560898132275, 24.969680218434, 23.344979203827982, 14.163803663922758, 24.19289557026432, 19.294777084234205, 27.018874635032084, 21.453845257303577, 15.354771838459385, 19.260681767511098, 22.3256866884191, 25.16438113772411, 28.34504899090593, 16.97508413768447, 22.861265985866588, 22.198557487355533, 20.885912866669543, 21.124145037975868, 27.195323878921386, 26.510619179332572, 27.702084747962793, 27.064364931285674, 26.314839899899283, 26.304514967942048, 25.255484645871274, 29.097729861383232, 29.364083891702943, 23.53563667156225, 25.23140988751803, 27.23062102128833, 23.78259445631638, 21.749437581241992, 22.23150771719244, 25.412676461151726, 1.8.361285776688067, 26.120772013695362, 23.64444086682674, 21.916986986308334, 24.689318611957763, 30.899456039632174, 27.578146270324023, 19.69904101478267, 25.480671083929604, 16.925487465864002, 20.570869967541608, 24.88585211613103, 21.301827336243225, 20.271868888594003, 25.273955878059, 26.124268366637907, 18.89779677220731, 22.921143451762415, 21.34280341274386, 23.69778718214303, 22.297886941487175, 13.70254237152109, 22.881061934509418, 18.570521329041096, 25.66879599334509, 20.461415451887618, 14.651228917990927, 18.414117293601524, 21.415427425986707, 23.836651627009054, 26.830055759027662, 16.272866744896156, 21.781728765043493, 21.125975396105574, 19.941161195211794, 20.4703153576848, 25.654769116912682, 25.06730302017108, 26.15126622364446, 25.500254847433485, 24.67370995890158, 25.013962810039903, 23.994532240980046, 27.61598423670196, 27.62868277749768, 22.344091798628963, 23.932952007139935, 25.700755130933455, 22.34890084238052, 20.70853108443423, 21.103375830794207, 23.994398394704877, 17.480117394940393, 24.67642116611427, 22.40794952339445, 20.896731509710854, 23.35711813223298, 28.87960799934916, 25.958243070264484, 18.646724051144165, 24.065012943551267, 16.151286908759488, 19.78126904990392, 23.741079266585725, 20.322146364574607, 19.213459295558053, 24.02699618757554, 24.843082844595777, 18.107779046429492, 21.827044632361975, 20.46937940228014, 22.499436123529115, 21.308916679661028, 13.259798455268516, 21.64787950170155, 17.88174550770157, 24.394684450536012, 19.520557054274587, 13.984979070911425, 17.61358549396299, 20.55480217723247, 22.586151395688276, 25.404706304777587, 15.611919245956695, 20.76276945078049, 20.111075959894606, 19.047907680477493, 19.0283351247552, 2.4.21126825868889, 23.70939953205418, 24.700803778567817, 24.04124436910061, 23.138611732473105, 23.803921332656238, 22.808877063992348, 26.22691067225861, 26.006971606656098, 21.223368492480994, 2.2.708269087301158, 24.268159596395044, 21.007621682079385, 19.722984236369786, 20.043980891220485, 22.663742918069985, 16.644867799775664, 23.320762876962647, 21.24587386864088, 19.932003133921583, 22.10086271476073, 27.00489887326641, 24.442658440030613, 17.657825164256572, 22.736955801981303, 15.41813455736283, 19.02522059715989, 22.65912245395119, 19.39791322216006, 18.213614069693495, 22.848990798306012, 23.63713753504958, 17.357360141907474, 20.79134086633968, 19.64244564002136, 21.370060480753633, 20.374502493932376, 12.834777750273574, 20.48841472532017, 17.226550403977086, 23.191858073311632, 18.62837184028611, 13.353948499928867, 16.856378290277945, 19.74077829741002, 21.408126744574663, 24.06334553801667, 14.98961695060905, 19.800715166377625, 19.150541878257656, 18.2031043854274, 18.064751175662124, 22.85829071060632, 22.431544321369802, 23.343770472651922, 2.2.679964280969614, 21.702472666743457, 22.668748112725986, 21.693495238708707, 24.92402962879011,

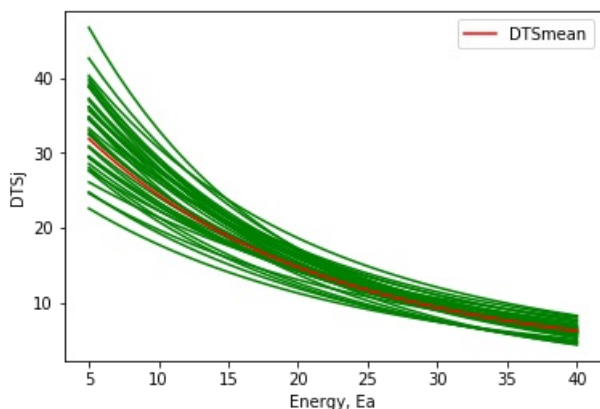
24.49096472261809, 20.16893431809337, 21.552913439682932, 22.926177353698925, 19.752499653883916, 18.789618960262686, 19.048580467974546, 21.414860902173317, 15.852980179917658, 22.048036619744497, 20.153309679530782, 19.019513640148997, 20.91601879107666, 25.264287334660843, 23.024218802388255, 16.728212818009524, 21.49070696906025, 14.72367496842968, 18.30118309976714, 21.636212603954988, 18.525693740620845, 17.26893896772196, 21.735900434499587, 22.501586590635885, 16.644333951411966, 19.810715705218342, 18.859288851147813, 20.30539173456906, 19.491309723751225, 12.426722835813061, 19.39805211403821, 16.60314575453028, 22.055943158824032, 17.78213328659321, 12.756183103950747, 16.13995553746679, 18.97052136011666, 20.298127592886246, 22.800692861314694, 14.403511133256858, 18.892126414960654, 18.241256195948946, 17.403892036914677, 17.15316767590114, 21.589765536804343, 21.228725654794882, 22.073736179863687, 21.409580552734965, 20.35870541279478, 21.603233534403923, 20.643725281806628, 23.70134804587651, 23.07326513330491, 19.176562926183276, 20.46271614039187, 21.668631774875482, 18.577716740192002, 17.90545035004263, 18.11278868768744, 20.242317505391632, 15.102048080607593, 20.8528711857508, 19.12569710573454, 18.15618058510903, 19.798334249411212, 23.64760442085355, 21.696269167202775, 15.854038757034104, 20.320874869201383, 14.065695365659597, 17.607689971209684, 20.6688209429503, 17.702277878356604, 16.37624373258009, 20.683932355787803, 21.43192103300862, 15.966628977142772, 18.88204989022351, 18.117362484322207, 19.301439215358123, 18.65621942167053, 12.034911232158024, 18.372473308382755, 16.00984368385679, 20.98285295506402, 16.979276012258808, 12.189841338881374, 15.461934174039078, 18.241381578882386, 19.2519875502171, 21.61181635709677, 13.85131682121801, 18.033781784631124, 17.38028971631504, 16.647587988597643, 16.29065374720387, 20.400047981188223, 20.09626043223739, 20.88473097905688, 20.22375431172156, 19.101173719055886, 20.602565167816685, 19.655240466300807, 22.55332098632744, 21.747019579174614, 18.242312440954905, 19.43376905172706, 20.48979055607157, 17.47782519881523, 17.06767440134413, 17.232547797919032, 19.141061449317835, 14.38980614302782, 19.730257528719424, 18.158795298606538, 17.3391138795968, 18.74382035399452, 22.145482654630026, 20.45263412584496, 15.031717888902731, 19.222440265984773, 13.442116633233509, 16.943345759102854, 19.753639908209635, 16.924664574335342, 15.532529480891263, 19.68952477782436, 20.423943900349844, 15.322299449044042, 18.002409200469604, 17.414276134426743, 18.354471318837803, 17.86631387671199, 11.658653690285403, 17.40763755674881, 15.44505258738067, 19.968767903012935, 16.21738588801612, 11.653187517764804, 14.820078098199302, 17.550881197618065, 18.265805325490962, 20.49210881560365, 13.330901451967767, 17.22266367768073, 16.564889232343194, 15.931674969945457, 15.474448860069815, 19.28388849695595, 19.02977184337982, 19.771211301595635, 19.11660488874076, 17.92416077181354, 19.6622952476749, 18.724023378363903, 21.474816414039903, 20.505877125484727, 17.36250543598697, 18.462408030451513, 19.384332407565985, 16.447904801043514, 16.273656549312605, 16.40410209160066, 18.106396657012542, 13.714121437310927, 18.675523554042925, 17.248658982745553, 16.56560327209546, 17.748734861746915, 20.749291054256897, 19.28758187555284, 14.257909631939983, 18.19072961120749, 12.850984897136152, 16.306822553373564, 18.887574865444535, 16.19004765128328, 14.73497688812572, 18.749332308879602, 19.47374731765458, 14.709517069869493, 17.169033071778383, 16.74778565119847, 17.460998029814732, 17.118863136040435, 11.297292569496541, 16.499763495838465, 14.907271166033984, 19.010117288720895, 15.49419077010084, 11.144585522394078, 14.212288719649031, 16.896702768325472, 17.335927381515223, 19.43726546850693, 12.840274336320752, 16.45594499423218, 15.792466520692997, 15.253790569006929, 14.701952702115005, 18.23640409128164, 18.025168587195093, 18.728028624214712, 18.082675695622218, 16.822339804981116, 18.778310969834344, 17.846342488309975, 20.461082842288924, 19.343950985709498, 16.533710377184764, 17.54519724269478, 18.347316322774642, 15.483162417870972, 15.520920958653727, 15.623973990716841, 17.133956066769855, 13.072985349745426, 17.6843107101235, 16.391616816544605, 15.83310667467496, 16.80956625484099, 19.451075533689842, 18.195791032013577, 13.529500619765352, 17.22139035774885, 12.290463654849036, 15.696856581053272, 18.06772385581413, 15.495802694192799, 13.980935120060266, 17.860212334181153, 18.57769133520909, 14.126563340167495, 16.37932393609922, 16.1157838885532, 16.61775466070693, 16.41131245457773, 10.950200298239524, 15.645312147119506, 14.395083399338013, 18.10356220188098, 14.807551818299547, 10.662492900330276, 13.63659614057587, 16.27667825820592, 16.45893175010091, 18.44326330319495, 12.377576869148525, 15.730976705460494, 15.060588047773809, 14.611717400154381, 13.970715660083734, 17.253051807844898, 17.07862554260512, 17.750400512083615, 17.116902712248276, 15.790746813530415, 17.946807350436302, 17.018730571502555, 19.507719611976793, 18.255783336227307, 15.75272442126758, 16.67891451108739, 17.374153221424244, 14.579280029143279, 14.807140515130916, 14.888942105335444, 16.219677449571822, 12.464505987482735, 16.752552250106284, 15.584251400619033, 15.139239272887604, 15.9230190133121, 18.243504237429, 17.172320008152568, 12.843588661905953, 16.310368088002324, 11.75882641670165, 15.112244977785341, 17.291368005834123, 14.839474835373357, 13.267911462510481, 17.019212284497115, 17.7323843935021, 13.571822420026592, 15.630837234813294, 15.516292052407046, 15.821686718332192, 15.741270607240065, 10.616777913692431, 14.840971046966015, 13.90715308160913, 17.2459797052277, 14.155453605316, 10.205455322975515, 13.091150922610323, 15.688778923843168, 15.631612928615272, 17.506341842136806, 11.941073433530212, 15.045276256006952, 14.36696534041988, 14.00337391131162, 13.278429877793116, 16.32960418376843, 16.186565786599562, 16.83388383224569, 16.21458537872076, 14.824755217261616, 17.164262416231217, 16.237964828638688, 18.610649579961844, 17.236312865628342, 15.016557467034767, 15.860537624818864, 16.460579778674393, 13.732204768165145, 14.130127471227624, 14.196021100064302, 15.35978107230059, 11.886901066554525, 15.876453042046924, 14.823380804444124, 14.4817633654932, 15.085999857178662, 17.119817393601377, 16.21257875520092, 12.19746786740541, 15.453885321047363, 11.254449824682105, 14.551842726681539, 16.555958913805355, 14.218767382192384, 12.593561604447602, 16.223557729830205, 16.93466528453857, 13.043774487793954, 14.921272062292134, 14.947451607619444, 15.069935818747037, 15.10649900316756, 10.296453675910005, 14.083639433790271, 13.442219252381506, 16.43444812625344, 13.536005268758222, 9.772101381930007, 12.574216399584268, 15.131105897228561, 14.850967784319216, 16.622985280403167, 11.529142948212769, 14.396516739347032, 13.70944597696086, 13.426805788247025, 12.622920854629545, 15.462126530291831, 15.345643863678667, 15.974349971604108, 15.371359704639564, 13.920052332934047, 16.427414516350176, 15.501048567390486, 17.766094016385345, 16.280844851532493, 14.32241736471464, 15.087231548048857, 15.60263426720068, 12.93816460547362, 13.487824701955494, 13.54244321457468, 14.550749061527021, 11.338491251338473, 15.052470814648622.



14.106041492026232, 13.858578883727192, 14.295604892441075, 16.073781304566385,  
15.312302678693458, 11.588614845363793, 14.648421872259975, 10.775807216846824,  
14.014559755666726, 15.859107493280781, 13.631531228218345, 11.955680532543013,  
15.470641243154214, 16.181586490259573, 12.540989559076541, 14.248462398962998, 14.40751670730661,  
14.35982657573946, 14.504901545657733, 9.988681752567405, 13.370414420730395, 12.999091689541515,  
15.66623338935535, 12.947415813324453, 9.361137702343132, 12.084161498755424, 14.601881432174737,  
14.114182397940931, 15.789905882428545, 11.140271011940564, 13.782516793734887,  
13.086005157503196, 12.880177916373935, 12.002139551417601, 14.646955898260316,  
14.552730217471009, 15.167961906518345, 14.583173422489319, 13.072617523278224,  
15.733241565363352, 14.805194318853978, 16.970549527076876, 15.385023560397707,  
13.667696196108745, 14.356336466509914, 14.796634253230366, 12.193648296195017,  
12.878297529823492, 12.925641298113867, 13.789306334394615, 10.817693915776612,  
14.277298734095858, 13.429472537053881, 13.267714543213168, 13.549107599886387,  
15.099646126950184, 14.467528557256852, 11.014675902817329, 13.890696648762773, 10.32146260764427,  
13.499358184927425, 15.19857356907414, 13.07554992713897, 11.352193997920697, 14.758011982901921,  
15.470398788596128, 12.062121732181177, 13.610368895821091, 13.894847110222745,  
13.688854393968672, 13.934515185541473, 9.692940970553211, 12.698578087349151, 12.576646716959084,  
14.938776312639437, 12.388004964996506, 8.971344353349275, 11.61945403578934, 14.099440763464733,  
13.418619782678665, 15.004028546493297, 10.773042601622222, 13.201231169940863,  
12.494737815045358, 12.361766863165784, 11.414154972544928, 13.88068160032156, 13.80489670184985,  
14.411152982949309, 13.846263025924987, 12.278701901120064, 15.078942045159021,  
14.14780827254556, 16.220766832787355, 14.54480678637059, 13.049957545054657, 13.665356616241981,  
14.039156000352918, 11.495386793682952, 12.299726080803993, 12.343233230043618,  
13.072402973855539, 10.323017298665306, 13.54784921526892, 12.791101026801929, 12.70731958525623,  
12.84394760985923, 14.192107122790343, 13.674572305995332, 10.473455166346483, 13.177650773059248,  
9.890065056491663, 13.005249716530688, 14.572256173894262, 12.549555837355532, 10.781150518453963,  
14.083365946424141, 14.798537026679046, 11.6059038287661, 13.00507117483622, 13.407901554189273, 1  
3.054674102737588, 13.393501119473793, 9.408733630869598, 12.065585428350998, 12.173823270472495,  
14.249680799257453, 11.856188115970799, 8.60157053696778, 11.178654451227134, 13.622224534598162,  
12.761808419233757, 14.26247645223186, 10.426135284457647, 12.650741925342858, 11.933851231660935,  
11.869953846776836, 10.857147195159943, 13.16012717168169, 13.099403094866805, 13.700607277027665,  
13.157132546420288, 11.534809475832136, 14.461917609175405, 13.526475924970663,  
15.513731251307696, 13.756442359608775, 12.466924683992447, 13.011949842217586,  
13.32701546704115, 10.840336020065363, 11.750398135754496, 11.793007609492465,  
12.397197935372633, 9.853055027210251, 12.861238877794227, 12.188528562159519, 12.175656066895572,  
12.177720210093606, 13.346269090664869, 12.930008439076634, 9.962903559326085, 12.506431935441736,  
9.480343398827294, 12.531293158705916, 13.978184497334103, 12.051170912701481, 10.240713882330349,  
13.444536849905315, 14.163606968961185, 11.17114240088183, 12.43076061204108, 12.94523155568237,  
12.455089371008983, 12.880136588074281, 9.135584383491949, 11.469053101561459, 11.789619205857987,  
13.596702858371508, 11.350472193010074, 8.25073053804909, 10.760409958429193, 13.168771753247334,  
12.141431551574325, 13.562557712781421, 10.098312907068198, 12.129250202045675,  
11.401658126413142, 11.403218158772708, 10.329400818070978, 12.482333660438531,  
12.433684544374865, 13.033241415995214, 12.5125339333879, 10.83767963763901, 13.879757146987307, 1  
2.938948839739266, 14.846644741599388, 13.016446468195742, 11.916469608178978, 12.393917838170033,  
12.657250631934394, 10.225660892629643, 11.228702444192326, 11.272910607291902,  
11.761043981250085, 9.406480984784025, 12.214774564015318, 11.619518768582932, 11.671091662271708,  
11.548166537354081, 12.557613710514644, 12.230651097859882, 9.481108571834737, 11.874379883365236,  
9.091101315613471, 12.076592077685383, 13.414509442329322, 11.578949390311424, 9.729156120838015,  
12.839487593256147, 13.563373135550002, 10.756713077791364, 11.885733572271356, 12.50547560767509,  
11.88804284856109, 12.392807232063461, 8.873039159023437, 10.906748921356877, 11.423087833473737,  
12.977740395664847, 10.869450137616475, 7.917799918971842, 10.363449075083643, 12.737713236543158,  
11.555317191941986, 12.901752959011834, 9.788419727376631, 11.635068549600236, 10.896570183924528,  
10.960131011030654, 9.829298804673565, 11.844544146952627, 11.805339879357938, 12.406187748751217,  
11.909448913581256, 10.184270882994893, 13.30222180339176, 12.38313243696711, 14.216909381000278,  
12.32158365033856, 11.396602854391897, 11.809197022244808, 12.027105460298063, 9.648720512637011,  
10.733122469587283, 10.781033882565517, 11.161473747064097, 8.982044500446431, 11.605940342099386,  
11.081985739593641, 11.192092940333211, 10.953164407031673, 11.82196955855351, 11.573536521679221,  
9.026284764957476, 11.279012962989391, 8.721212718896465, 11.640292570380073, 12.879495747345741,  
11.131345039559655, 9.244850920409784, 12.266302271328206, 12.995747552955715, 10.361556227994702,  
11.368385066564286, 12.087353749683496, 11.351606982148912, 11.929999967519835, 8.62066415415311,  
10.376582047403662, 11.073334665256318, 12.39082371777452, 10.411795729789178, 7.601811943843923,  
9.986576512317255, 12.32776551109334, 11.001428787159373, 12.27770378866158, 9.495374957485936,  
11.16661375559156, 10.41709199462911, 10.539349778878252, 9.355316695824133, 11.244189399491603, 1  
1.212120725750855, 11.816778763788163, 11.34507221542339, 9.571745691368568, 12.811233473561233, 1  
1.857074722454099, 13.622112157474835, 11.668848325078534, 10.905464045839429, 11.255850006173935,  
11.43401469531466, 9.107054430228507, 10.262230537446664, 10.315603474734152, 10.596186852294306,  
8.57856583298493, 11.03285423163069, 10.573983340701137, 10.73721908640384, 10.390719737995543,  
11.135484569384134, 10.955906848282316, 8.596764955573155, 10.718015634555917, 8.369617432547924,  
11.221581151528223, 12.371514635476306, 10.706909311094856, 8.78626744590644, 11.723178695420971,  
12.458779345788406, 9.984672913755084, 10.87720280512379, 11.689662485694463, 10.843975459216313,  
11.490296344145161, 8.378044868089212, 9.876593821984162, 10.739514360671578, 11.83410669914235,  
9.97625873344311, 7.301854217939312, 9.628668397254854, 11.937725135208963, 10.477856501593319, 11  
.688202018179384, 9.218167688123605, 10.722400149875554, 9.961815379691732, 10.139612614411037, 8  
.906017170070061, 10.67887458054483, 10.651921369950617, 11.262532660182641, 10.816796052948765, 8  
.997456470357273, 12.320858752026684, 11.358955885597688, 13.060010968842223, 11.055447740340801, 1  
0.441313109477749, 10.732057618539757, 10.875590089823866, 8.598369905612634, 9.814682359428486, 9  
.874969589360663. 10.063037974264054. 8.194931974146549. 10.491912926543622. 10.093695307376821. 1

0.305116037397653, 9.858958532935564, 10.49460074137163, 10.375195139996961, 8.19099203174568, 10.189226889231836, 8.035317148725593, 10.819682749294381, 11.889036954341607, 10.304284891602014, 8.351964548938165, 11.208421391495797, 11.950645103736102, 9.625121117147668, 10.410761620533593, 11.31127002726533, 10.363455235113763, 11.072366352988782, 8.144785187293603, 9.40494921236822, 10.420827859085255, 11.305858563589004, 9.561660342993072, 7.0170655290165005, 9.288667806553052, 11.566463413197612, 9.982809075196078, 11.13117967966893, 8.955852167363473, 10.30103335058193, 9.529414075401391, 9.759733405674044, 8.480044930044311, 10.14636692480306, 10.122769317312434, 10.741139984570642, 10.322195772355977, 8.458932492146538, 11.857301432015072, 10.887078695341753, 12.528521730022424, 10.47878622687874, 10.002522116066828, 10.236111445363937, 10.34960756294325, 8.120530092750947, 9.389211908543905, 9.45759720324949, 9.560025810934013, 7.8300926202618495, 9.982469432190806, 9.639426075981833, 9.894511002579877, 9.356119377158002, 9.896030898628162, 9.82901154002818, 7.807511352560046, 9.690629500549173, 7.717371641906487, 10.433858803619417, 11.430626773209092, 9.92219969702943, 7.9405853367207, 10.7204350437838, 11.46963996347, 9.282012218238183, 9.967718237555982, 10.95111840589709, 9.908459102936352, 10.674962652459671, 7.920506515988415, 8.959928817652102, 10.116519686825436, 10.804456236437488, 9.166888911940367, 6.746632878018121, 8.965580589993607, 11.212921473785148, 9.514606217899637, 10.60469970958392, 8.707543408324156, 9.901204422176678, 9.11863875354144, 9.398597059018423, 8.076121895143567, 9.644584315876088, 9.622816496717329, 10.2504512537312, 9.859016571811066, 7.953867749975282, 11.418890274137983, 10.439859630809927, 12.02570649766305, 9.936450655605384, 9.587567697097171, 9.766406853743078, 9.853995342499239, 7.671543077321271, 8.984626622207351, 9.06205742059823, 9.085282864006507, 7.483056516239464, 9.502135263640024, 9.209592291398005, 9.504207343730902, 8.880546421332138, 9.336737338772323, 9.315130470936076, 7.444963689684459, 9.220340048645935, 7.414895223570826, 10.063405461926704, 10.994935406088366, 9.559461273771095, 7.550852078549269, 10.25771835459054, 11.014169349994525, 8.954507707459996, 9.546806367399018, 10.608186477727315, 9.47749876802599, 10.296915183661369, 7.704846950043551, 8.539921402245252, 9.825875427962963, 10.328377226268247, 8.790895945544149, 6.4897886874490425, 8.658471464670182, 10.87610568676817, 9.071671504292723, 10.106947279749424, 8.472413102383262, 9.521684417903572, 8.728312355818657, 9.055155083459404, 7.69304268183752, 9.171584694422776, 9.15033106581333, 9.78846548857396, 9.425161212169378, 7.4801096684876685, 11.004069880447503, 10.015820688264151, 11.549762529118398, 9.426197004011653, 9.195023996220858, 9.321436466566725, 9.386822997989213, 7.249551705875998, 8.599802911472569, 8.68701951782463, 8.63706597923108, 7.152887935094903, 9.04911544917698, 8.802714939562529, 9.13307979039038, 8.430682816043468, 8.813912210077504, 8.831478794367168, 7.102078671119876, 8.776599660195147, 7.127053421755108, 9.707651867077843, 10.580695831736481, 9.214951579454931, 7.181561428454935, 9.818858293068569, 10.582741326426383, 8.641816115604179, 9.146832105777776, 10.281551673504994, 9.069178391896997, 9.937126148132544, 7.49746049198303, 8.143416920753879, 9.54821934854078, 9.876192998398873, 8.432692340821948, 6.245808176475116, 8.366460361647876, 10.555083393887017, 8.652525734979385, 9.63622172491787, 8.2496858161506, 9.161319280810325, 8.357325721903429, 8.728421457274994, 7.329670354108686, 8.725556235529917, 8.703689774790899, 9.353319591015614, 9.018678642747133, 7.035648606688067, 10.611391962564541, 9.613581810796187, 11.099012199708062, 8.945937944910789, 8.82355611604482, 8.899784058509391, 8.946291295092722, 6.852824147856624, 8.23368195626266, 8.331243620011824, 8.213747585794085, 6.838703410793671, 8.621731312643304, 8.417412057196755, 8.780069967548753, 8.005064568201677, 8.324959474832527, 8.376124857561228, 6.777668690537882, 8.357765411228728, 6.853059870754954, 9.365958532744294, 10.186717483529335, 8.88762211699231, 6.83157994399329, 9.40252470756149, 10.173959505282555, 8.34319014607725, 8.766669615442032, 9.97032069081797, 8.682188572861456, 9.594565322975884, 7.298016304970639, 7.7690000014027305, 9.282912164641923, 9.44656280496594, 8.091344858209853, 6.014006892481326, 8.088719008224684, 10.248978931656932, 8.255780733315909, 9.190929024426747, 8.03863545202237, 8.819025079321493, 8.00463349200249, 8.417468758401458, 6.984932427602113, 8.304808237934335, 8.281370849573545, 8.943278500476323, 8.637753472219657, 6.618608096691199, 10.239507314705833, 9.23185389152882, 10.671893708543543, 8.493731377675154, 8.47191402411548, 8.500118845495734, 8.53072280822686, 6.4797451365074705, 7.885265767769863, 7.993573956771751, 7.8138075812636805, 6.539668666393049, 8.218412648885119, 8.052391974777775, 8.444182214723567, 7.602314791390925, 7.867478327813071, 7.947268358079925, 6.470623248325839, 7.962302344049702, 6.5921733972469845, 9.037715801618909, 9.811881384034969, 8.57648939740568, 6.49983988240897, 9.007465277828894, 9.786516478114052, 8.057923995224346, 8.40525705100158, 9.673658898200404, 8.315300732965085, 9.268265690121805, 7.106198003750861, 7.415343857364078, 9.029348945301713, 9.038227939062258, 7.765972810226695, 5.793738389478603, 7.824467730085635, 9.95696992551302, 7.8801335483606945, 8.769574798631583, 7.838581953600308, 8.493783554955469, 7.669250266160948, 8.121424541409592, 6.657817112076733, 7.907762672068917, 7.88194735810503, 8.55672607156451, 8.280696220192985, 6.227235765603565, 9.887158431419769, 8.869432305108779, 10.266952509028485, 8.067769828495724, 8.138926884661386, 8.121190141708306, 8.138553232673983, 6.128807838778281, 7.553613500474169, 7.672932649762624, 7.435825812779433, 6.254995730892395, 7.837690442134132, 7.706447052255216, 8.124479676816444, 7.221138324147178, 7.439247949803083, 7.543230963645545, 6.179903692970937, 7.588776053106048, 6.343695290020328, 8.722342382135599, 9.45513560087113, 8.280630708678876, 6.185335256628964, 8.632500785049547, 9.419187723769005, 7.785350847569411, 8.061592877843148, 9.39078056459824, 7.9673618830085005, 8.957319358110272, 6.921702980630087, 7.081204597437802, 8.78695714184534, 8.650006382765232, 7.455744953434134, 5.584392044356857, 7.57297245871947, 9.678283836122255, 7.524361036861585, 8.370757783660792, 7.648888238614528, 8.184637961305944, 7.350247003677898, 7.83946794499044, 6.3473697767699274, 7.532946337957116, 7.50408102601052, 8.192156618979874, 7.945934290028343, 5.859894891709527, 9.553172715221013, 8.52519092594542, 9.88283340545947, 7.666370652042781, 7.823497785184218, 7.761822359949371, 7.768323342436355, 5.79860630804381, 7.237837997414612, 7.36831398814768, 7.078475109053215, 5.983940232742399, 7.478190088653638, 7.3784478701909775, 7.820080648501997, 6.860316691911974, 7.038213486201088, 7.162447628978605, 5.90453833738516, 7.2358457980961335, 6.1069667413661035, 8.419283959596006, 9.115491002053112, 7.9991801694789935, 5.887118134667805, 8.276520678979162, 9.070825958731067, 7.524840533795899, 7.734732063269002, 9.120945857270069, 7.637289739436089, 8.660873756293828, 6.744241764686222, 6.7654159094591035, 8.555194735770288.

7.007209709100000, 0.000070700200000, 0.711211701000000, 0.700120009100000, 0.000010700700000,  
8.280787821096556, 7.159876570869368, 5.385391002550387, 7.333541930470894, 9.412194740103176,  
7.187314798956819, 7.993163750697187, 7.468957343243696, 7.890689174826226, 7.046747647126702,  
7.570826514946592, 6.052689631611842, 7.178983587709381, 7.146516470316391, 7.84816707977915,  
7.632003608130513, 5.515056549688655, 9.236456224194528, 8.198076595016522, 9.518273262022348,  
7.287966972280724, 7.524598829314875, 7.420910332791342, 7.418671544225015, 5.48782847692216,  
6.937102553450329, 7.078779151983207, 6.740514821313431, 5.725798858759717, 7.138625087798111,  
7.067337841918658, 7.530155155159124, 6.518703390054339, 6.662473149978982, 6.803458556118134,  
5.64361789774878, 6.9022581057112, 5.881366448972413, 8.12801187782037, 8.79201729054686,  
7.731325048088078, 5.60429516708377, 7.938478923016092, 8.740355895881951, 7.275797340187857,  
7.42382968531433, 8.863458029668465, 7.324068168736328, 8.378128082775582, 6.573537412492016,  
6.4668840916104156, 8.333548497796174, 7.929528995992989, 6.877626732953951, 5.196190245213403,  
7.105525064533842, 9.158020328678083, 6.867916443989235, 7.635559838449926, 7.298229762889084,  
7.611092059275575, 6.75792595650166, 7.31477322868051, 5.77292659203874, 6.844589570813971,  
6.808075822145707, 7.523449746944934, 7.337540879173664, 5.191292302881721, 8.935987914069454,  
7.887104000150553, 9.172094274933457, 6.9310993051300205, 7.241266569496792, 7.097414933462606,  
7.088326981899702, 5.195249650664523, 6.650617882275582, 6.803451346904958, 6.4207848346035306,  
5.479906967935307, 6.817791168242456, 6.772128215003979, 7.253921754525659, 6.195219466880605,  
6.310266356705778, 6.464901748931785, 5.396291306543217, 6.586840824315799, 5.6663083679170345,  
7.848021887646414, 8.483839299145473, 7.476302328265966, 5.336024328112683, 7.617390099211131,  
8.426769380701513, 7.037657959078733, 7.127904084493229, 8.61766078701888, 7.026742936724167,  
8.108329988705982, 6.409324928729437, 6.1845834084909805, 8.121532351175112, 7.595249376255625,  
6.608295725654777, 5.01627477050043, 6.888308508057948, 8.915119108972291, 6.565153164437787,  
7.296789269756514, 7.136180975536522, 7.3450520669167325, 6.483002539973643, 7.070623708099525,  
5.507278347980759, 6.5285639635864054, 6.4876537113765265, 7.216785531699655, 7.061276410662083, 4  
.887267403670505, 8.650814333284456, 7.591350937527197, 8.84319776255147, 6.59440781020969,  
6.97259775503735, 6.790358976824595, 6.776103149271983, 4.919726464537625, 6.37763927390903,  
6.541511316571174, 6.118200013861264, 5.245636350316087, 6.5145608185507, 6.491893432756499,  
6.990644544345001, 5.888849386974151, 5.979962807073464, 6.145506116405025, 5.161761749070233,  
6.2884975985403395, 5.4612396030435555, 7.5788329587953065, 8.190133528091033, 7.233395505049738,  
5.0815118570285165, 7.3123257564525455, 8.129120876365443, 6.809889570618595, 6.846301104057468,  
8.382935818120474, 6.744417741678022, 7.850772482778442, 6.251350715159171, 5.917551750359682,  
7.918685832795653, 7.277027121191804, 6.351222635395016, 4.845157882011355, 6.68131433734654,  
8.682887793776782, 6.278073597463686, 6.975766425360437, 6.982319134838822, 7.09182206069939,  
6.221242068632316, 6.837733608722948, 5.254987593279093, 6.229785147067832, 6.184212588170215,  
6.927037715237539, 6.8020274638356675, 4.6017344658596295, 8.38004473321818, 7.309953924736273, 8.  
530558432016608, 6.27662512306866, 6.717745372893849, 6.498823382109048, 6.4808919724893075,  
4.6601912713449245, 6.117463930268263, 6.292193202143522, 5.831745051979413, 5.022393121784195,  
6.2278781934891025, 6.225766828814272, 6.7396303622847, 5.5986371565309225, 5.670052439429561,  
5.844085082867742, 4.939283029719572, 6.006202733228935, 5.265638432653188, 7.319986151808757,  
7.910124909076869, 7.001931594688107, 4.840009387142747, 7.022410986170534, 7.84652327144758,  
6.5919880468755245, 6.578224150584193, 8.158700482646637, 6.476250511790124, 7.604791040893631,  
6.099372046491386, 5.664886575422144, 7.724572646024665, 6.973995317281062, 6.105783080887295,  
4.682379577895554, 6.4839979048738865, 8.460758866611265, 6.00578395495894, 6.6714722498593755, 6.  
836182920976711, 6.850699342731438, 5.97195066342705, 6.615496173584127, 5.015339424519239,  
5.9472048003622, 5.896778358058135, 6.6531461535289145, 6.558692091256167, 4.3335275755636875,  
8.122846559075862, 7.042104138119204, 8.233219084389743, 5.976569723152279, 6.475914960244981,  
6.221943580308161, 6.201658326258688, 4.415646927751076, 5.869428467269816, 6.054780720667345,  
5.560469689600961, 4.809615746158997, 5.956754369558286, 5.9729366299130335, 6.500226165346941,  
5.323682693588996, 5.379136181287447, 5.559530666598727, 4.728156171608092, 5.738996418466205,  
5.0790124550469, 7.071043546935992, 7.643083780392315, 6.781278344016485, 4.61081125066081,  
6.7468212109429, 7.578143985998026, 6.383476269938092, 6.322965175079375, 7.94440564392622,  
6.22144994876795, 7.369760907011257, 5.953156571781384, 5.425741116389009, 7.538779299616412,  
6.685338468693336, 5.871387082691827, 4.5275050345164845, 6.295845822546616, 8.248198309870391,  
5.7474444042795385, 6.382949966623797, 6.697339538214106, 6.6210228753337415, 5.734473443296331,  
6.403339941273455, 4.787658892890829]



In [83]:

```
print(trainDf)
```

```
[
[0.7063304 0.70683079 0.00146251 0.00380824 0.03810557 0.00435117]
[0.70635824 0.70685776 0.0009269 0.00436612 0.03699839 0.00461898]
[0.7062753 0.70677508 0.0023524 0.00399443 0.04011978 0.00362131]
[0.70507969 0.70816283 0.0007991 0.00373872 0.03647169 0.00481945]
[0.70508043 0.70816376 0.00112761 0.00441216 0.03643647 0.00416218]
[0.70515163 0.70822705 0.00116249 0.00470674 0.03364698 0.00447271]
[0.70497209 0.70805397 0.00213385 0.00448844 0.04047727 0.00329023]
[0.70512345 0.70821216 0.00108097 0.00372102 0.03469957 0.00416813]
[0.70499394 0.70806682 0.00229394 0.00499382 0.03976673 0.00365116]
[0.70510936 0.708201 0.00156132 0.00390601 0.0351626 0.00424617]
[0.70507787 0.70816808 0.00194399 0.004127 0.03639642 0.0042015 ]
[0.70505221 0.7081249 0.00165128 0.00490711 0.03766153 0.00391327]
[0.70507852 0.70817095 0.00165183 0.00412144 0.03638661 0.00380655]
[0.7051117 0.70819071 0.00136318 0.00435519 0.03531069 0.0039676 ]
[0.70505706 0.70812225 0.00191622 0.00580065 0.03745558 0.00414509]
[0.70511098 0.7081959 0.0022079 0.00444033 0.03514236 0.00418274]
[0.7050988 0.70818457 0.00119817 0.0043708 0.03562715 0.00452358]
[0.70508355 0.70815602 0.00121567 0.00493547 0.03645463 0.00417974]
[0.70505706 0.70816132 0.00130334 0.00296242 0.03702264 0.00457704]
[0.70508309 0.70817155 0.00178777 0.00387336 0.03627496 0.00410139]
[0.70503467 0.70809967 0.00125189 0.00511298 0.03845587 0.00379598]
[0.70508214 0.7081516 0.00176524 0.00490285 0.03658378 0.00386822]
[0.70502216 0.70807892 0.00247448 0.00585524 0.03891895 0.00359186]
[0.70504112 0.70810271 0.00190288 0.0054176 0.0382351 0.00356104]
[0.70508284 0.70814928 0.00223183 0.00522234 0.03652303 0.00407387]
[0.70499943 0.70805743 0.00222693 0.00543951 0.03980151 0.00343145]
[0.70508513 0.70815153 0.00188308 0.00505022 0.03646822 0.00416835]
[0.70512912 0.70819857 0.00134921 0.00494838 0.03469615 0.00419405]
[0.70512149 0.70818669 0.00121321 0.00568853 0.03491568 0.00472959]
[0.70496588 0.70803683 0.0026339 0.00523145 0.04071138 0.00390173]
[0.70512088 0.70818279 0.00154685 0.00546709 0.0351007 0.00416293]
[0.70521402 0.70828029 0.00119675 0.00536464 0.03103101 0.00436812]
[0.70514399 0.70821334 0.00199449 0.00498158 0.03407711 0.00397269]
[0.70517178 0.70821584 0.00159418 0.0064473 0.03317127 0.00431863]
[0.70517706 0.7082317 0.0011725 0.00573147 0.03283775 0.00454287]
[0.70514669 0.70827163 0.00152096 0.00496716 0.03280142 0.00406623]
[0.7051332 0.70825049 0.00200978 0.00502252 0.03355059 0.00367569]
[0.70519527 0.70829508 0.00136221 0.00630411 0.0309362 0.00436683]
[0.70512085 0.70824255 0.00222259 0.0057106 0.03385778 0.0036305 ]
[0.70503297 0.70815625 0.00207109 0.00527798 0.03738018 0.00371634]
[0.70517821 0.70831469 0.00177091 0.00467644 0.03109363 0.004709 ]
[0.70508561 0.70822079 0.00220987 0.00444502 0.03520472 0.00368669]
[0.7050975 0.70821748 0.00111926 0.0055715 0.0348325 0.00446386]
[0.70510334 0.70822598 0.00228628 0.00529519 0.03455335 0.00424683]
[0.70552172 0.70759004 0.00192358 0.00508722 0.03883532 0.004204 ]
[0.70549668 0.70756591 0.00187139 0.00544231 0.0396964 0.00399867]
[0.70555045 0.70762789 0.00098409 0.00494446 0.03759683 0.00473215]]
]
```

### Problem 2-3: (11pts)

Using the same  $SE_{\{a\}}$  values and calculated  $DTS_{\{mean\}}$  from 2-2, predict the bloom date  $BD_{\{j\}}$  for each of the training years. Find the mean squared error relative to the actual  $BD$  and plot it against  $SE_{\{a\}}$ . Find the optimal  $SE_{\{a\}}^*$  that minimizes that error on the training data.

In [8]:

```
def mean_squared(y, predicted):
    sq_diff = np.square(np.subtract(y, predicted))
    err = np.sum(sq_diff)
    return err/len(y)

years = train_data["year"].unique()
error = []

for d in DTSmean:
    prediction = []
    actual = []
    dtsmean = np.round(d)
    for year in years:
        df = train_data.loc[(train_data["year"]==year)]
        df.reset_index(inplace=True)
        row = Dj[year-1961]
        counter = 0
```

```

selected_df = df.loc[row:,:]
for u, val in selected_df.iterrows():
    Eatemp=(DTSmean.index(d)+5)
    Tij = val["avg_temp"]+273.15
    ts = np.exp((Eatemp*(Tij-Ts))/(R*Tij*Ts))
    counter+=ts
    if counter>=d:
        prediction.append(doy(val["year"],val["month"],val["day"]))
        act=selected_df.loc[(selected_df["bloom"]==1)]
        actual.append(doy(int(act["year"]),int(act["month"]),int(act["day"])))
        break
error.append(mean_squared_error(actual, prediction))

print(error)
x_values = [x for x in range(5,41)]
plt.plot(x_values,error,label = "Ea VS error plot")
min_error = min(error)
idx = error.index(min_error)+5 # since the index starts from 0 but Ea value starts from 5

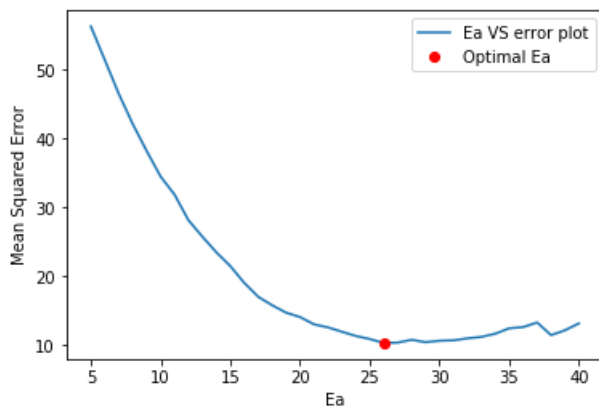
print("Minimum Error: ",min_error)
print("Optimal Ea: ",idx)
plt.plot(idx,min_error,"ro",label = "Optimal Ea")
plt.legend()
plt.xlabel("Ea")
plt.ylabel("Mean Squared Error")
plt.show()

```

```

[56.30769230769231, 51.42307692307692, 46.51923076923077, 42.13461538461539, 38.21153846153846,
34.5, 31.865384615384617, 28.134615384615383, 25.75, 23.48076923076923, 21.48076923076923,
19.03846153846154, 17.01923076923077, 15.788461538461538, 14.711538461538462, 14.076923076923077,
13.01923076923077, 12.576923076923077, 11.942307692307692, 11.326923076923077, 10.884615384615385,
10.307692307692308, 10.365384615384615, 10.76923076923077, 10.423076923076923, 10.634615384615385,
10.692307692307692, 10.98076923076923, 11.192307692307692, 11.653846153846153, 12.442307692307692,
12.615384615384615, 13.288461538461538, 11.450980392156863, 12.156862745098039,
13.156862745098039]
Minimum Error: 10.307692307692308
Optimal Ea: 26

```



## Problem 2-4: (4pts)

Using the  $BD_j$  dates from problem 2-1, the average  $DTS_{\text{mean}}$  from 2-2, and the best-fit  $E_{\text{a}}^{*}$  from 2-3, predict the bloom-dates  $BD_j$  for the years in the test set. Determine the error between your predicted  $BD_j$  values and the actual values, and evaluate this model using the coefficient of determination ( $R^2$  score).

In [24]:

```

test_data = pd.read_csv("test.csv")
years = test_data["year"].unique()
Ea = 26
dtsmean = np.round(DTSmean[Ea-5])

prediction = []
actual = []

for year in years:
    df = test_data.loc[(test_data["year"]==year)]
    df.reset_index(inplace=True)
    row = df[year-1961]

```

```

row = 0
year = 1901
counter = 0
selected_df = df.loc[row,: ]
for u, val in selected_df.iterrows():
    Tij = val["avg_temp"] + 273.15
    ts = np.exp((Ea * (Tij - Ts)) / (R * Tij * Ts))
    counter += ts
    if counter >= dtsmean:
        prediction.append(int(doy(val["year"], val["month"], val["day"])))

        act = selected_df.loc[(selected_df["bloom"] == 1)]
        actual.append(doy(int(act["year"]), int(act["month"]), int(act["day"])))
        break

actualValueForDTS = actual
predictedValueForDTS = predicted
print(actual)
print(prediction)
print(r2_score(actual, prediction))

[79, 89, 93, 90, 82]
[80, 88, 91, 90, 83]
0.9489795918367347

```

### Problem 2-5: (extra 10pts)

Discuss any improvements you could make to the model outlined above. If you have a suggestion in particular, describe it. How much do you think the accuracy would be improved?

1. According to the mentioned model and datasets, there is an assumption that the average temperature of a single day is sustained throughout the day. This is not the case in real life as the temperature can fluctuate even from hour to hour. So if the fluctuations of the temperature could be added, the accuracy and could possibly be improved.

2. There are different species of cherry trees from the Prunus genus. Different species of trees might have different type of implications to different changes of variables. We did not take that into account. If we can take those things into consideration, accuracy might be improved.

In the above proposed model we use only the temperature records of the past 61 years. However blooming date sakura depends on many other things as well like the rain fall and humidity of the year. So if we include those as a feature there is a chance that the accuracy may increase. There is no mathematical formula yet to include all these parameters as feature so machine learning approach could help us in this. We can try some popular ML algorithms to see what happens.

## 3. Predicting Bloom-date via Neural Network (30pts total)

### Problem 3-1: (20pts)

Build a neural network and train it on the data from the training years. Use this model to predict the bloom-dates for each year in the test set. Evaluate the error between predicted dates and actual dates using the coefficient of determination (R2 score). Only use the weather data given in `tokyo.csv` and the sakura data acquired in problem 0-1. You may use whichever framework or strategy that you like to construct the network.

In [12]:

```

fullDf = pd.read_csv('tokyo.csv')
years = fullDf['year'].unique()
#assuming the data for the first 3 months can be acquired or predicted
JanToMar = (fullDf['month'] >= 1) & (fullDf['month'] <= 3)

#calculate the bloom dates of all years
bloom = (fullDf['bloom'] == 1)
bloomDateDf = fullDf[bloom]

#when it will bloom starting from day 1 from a year
bloom = []
for index, row in bloomDateDf.iterrows():

```

```

    bloomDate=(dt.date(int(row['year']),int(row['month']),int(row['day'])) - dt.date(int(row['year']),1,1)).days+1
    bloom.append(bloomDate)

#split train,test and validation data
trainYear = (fullDf['year']!=1966) & (fullDf['year']!=1971) & (fullDf['year']!=1985) & (fullDf['year']!=1994) & (fullDf['year']!=2008)
testYear=(fullDf['year']==1966) | (fullDf['year']==1971) | (fullDf['year']==1985) | (fullDf['year']==1994) | (fullDf['year']==2008)

#print(trainYear)
#print(testYear)

trainDf=fullDf[trainYear & JanToMar]
testDf=fullDf[testYear & JanToMar]

trainDf=trainDf.reset_index(drop=True)
testDf=testDf.reset_index(drop=True)

#keeping year,local_avg, sea_level, total_rainfall,avg_temp,avg_hum,sunshine_time,bloom
trainDf=trainDf.drop(["bloom","serial","month","day","max_rain_1_day","max_rain_10_min","highest_temp","lowest_temp","lowest_hum"], axis=1)
testDf=testDf.drop(["bloom","serial","month","day","max_rain_1_day","max_rain_10_min","highest_temp","lowest_temp","lowest_hum"], axis=1)

#validation years=1963,1973,1983,1993,2003 - manual arbitrary selection
selectedTrainYear = (trainDf['year']!=1963) & (trainDf['year']!=1973) & (trainDf['year']!=1983) & (trainDf['year']!=1993) & (trainDf['year']!=2003)
validYear=(trainDf['year']==1963) | (trainDf['year']==1973) | (trainDf['year']==1983) | (trainDf['year']==1993) | (trainDf['year']==2003)
validationDf=trainDf[validYear]
trainDf=trainDf[selectedTrainYear]

trainDf=trainDf.reset_index(drop=True)
validationDf=validationDf.reset_index(drop=True)

#sum of all features for each year
trainDf=trainDf.groupby('year').sum().reset_index()
testDf=testDf.groupby('year').sum().reset_index()
validationDf=validationDf.groupby('year').sum().reset_index()

trainDf=trainDf.drop(["year"], axis=1)
testDf=testDf.drop(["year"], axis=1)
validationDf=validationDf.drop(["year"], axis=1)

#normalization for equal effect of all features
trainDf=normalize(trainDf)
testDf=normalize(testDf)
validationDf=normalize(validationDf)

y_train=[]
y_valid=[]
y_test=[]
i=0
for i in range(len(bloom)):
    if i in [1966-1961,1971-1961,1985-1961,1994-1961,2008-1961]:
        y_test.append(bloom[i])
    elif i in [1963-1961,1973-1961,1983-1961,1993-1961,2003-1961]:
        y_valid.append(bloom[i])
    else:
        y_train.append(bloom[i])
#print(y_train)
#print(y_valid)
accuracyList=[]
maxAccuracy=0
predictionList=[]
for i in range(100):

    clf = MLPRegressor(hidden_layer_sizes=(7),activation="relu",solver="lbfgs",max_iter=5000)

    clf.fit(trainDf,y_train)
    accuracy=clf.score(validationDf,y_valid)
    if accuracy>maxAccuracy:

```

```

        maxAccuracy=accuracy
        #print (maxAccuracy)
        bestClf=clf
        accuracyList.append(accuracy)
        accuracy=bestClf.score(testDf,y_test)
        predictionList=bestClf.predict(testDf)
        #print (predictionList)
        print("Accuracy: ",accuracy)

actualValueForNN=y_test
predictedValueForNN=predictionList

```

Accuracy: 0.5361472282817926

In [85]:

```
trainDf.shape
```

Out[85]:

(47, 6)

### Problem 3-2: (10pts)

Compare the performance (via  $R^2$  score) of the 3 implementations above: the 600 Degree Rule, the DTS method, and the neural network approach. For all methods, and each test year, plot the predicted date vs. the actual date. Discuss the accuracy and differences of these 3 models.

In [25]:

```

r2sTmean= r2_score(actualValueForTmean,predictedValueForTmean)

r2sDTS= r2_score(actualValueForDTS,predictedValueForDTS)

r2sNN= r2_score(actualValueForNN,predictedValueForNN)

testYears=[1966,1971,1985,1994,2008]
print("R squared for Tmean: ",r2sTmean)
print("R squared for DTS: ",r2sDTS)
print("R squared for NN: ",r2sNN)

plt.figure(figsize = (20,5))
plt.subplot(131)

plt.plot(testYears,actualValueForTmean,color='g',label="Actual Value")
plt.plot(testYears,predictedValueForTmean,color='r',label="Predicted Value")
plt.ylabel("Date in number of days")
plt.xlabel("Year")
plt.title("Tmean Plot")
plt.legend()

plt.subplot(132)

plt.plot(testYears,actualValueForDTS,color='g',label="Actual Value")
plt.plot(testYears,predictedValueForDTS,color='r',label="Predicted Value")
plt.ylabel("Date in number of days")
plt.xlabel("Year")
plt.title("DTS Plot")
plt.legend()

plt.subplot(133)

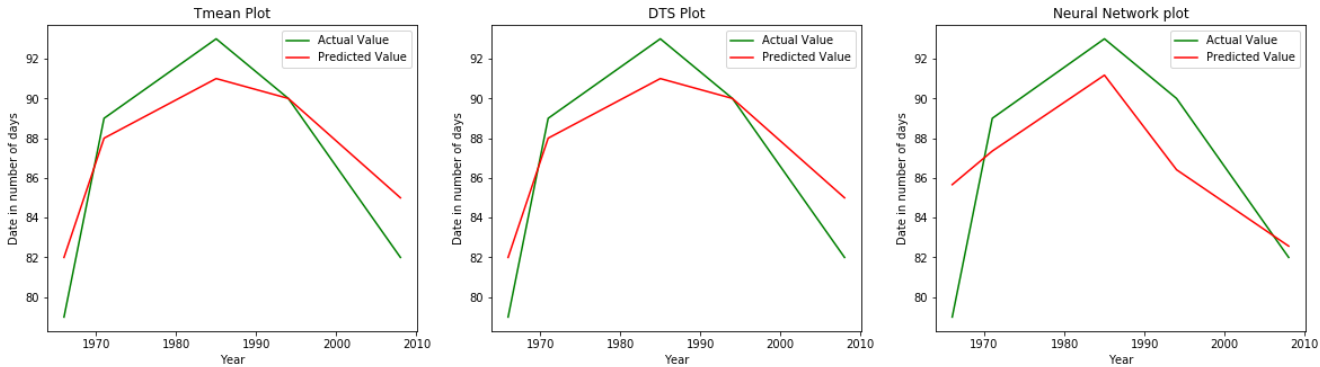
plt.plot(testYears,actualValueForNN,color='g',label="Actual Value")
plt.plot(testYears,predictedValueForNN,color='r',label="Predicted Value")
plt.ylabel("Date in number of days")
plt.xlabel("Year")
plt.title("Neural Network plot")
plt.legend()
plt.show()

```

R squared for Tmean: 0.8323615160349854  
R squared for DTS: 0.8323615160349854



R squared for NN: 0.5361472282817926



#### 4. Trends of the Sakura blooming phenomenon (20pts total)

##### Problem 4-1: (20pts)

Based on the data from the past 60 years, investigate and discuss trends in the sakura hibernation ( $\$D_{\{j\}}\$$ ) and blooming ( $\$BD_{\{j\}}\$$ ) phenomena in Tokyo.

In [44]:

```
BloomDate
```

Out[44]:

	serial	year	month	day
0	0	1961	1	1
1	1	1961	1	2
2	2	1961	1	3
3	3	1961	1	4
4	4	1961	1	5
5	5	1961	1	6
6	6	1961	1	7
7	7	1961	1	8
8	8	1961	1	9
9	9	1961	1	10
10	10	1961	1	11
11	11	1961	1	12
12	12	1961	1	13
13	13	1961	1	14
14	14	1961	1	15
15	15	1961	1	16
16	16	1961	1	17
17	17	1961	1	18
18	18	1961	1	19
19	19	1961	1	20
20	20	1961	1	21
21	21	1961	1	22
22	22	1961	1	23

23	serial	year	month	day
24	24	1961	1	25
25	25	1961	1	26
26	26	1961	1	27
27	27	1961	1	28
28	28	1961	1	29
29	29	1961	1	30
...	...	...	...	...
20514	20514	2017	3	2
20515	20515	2017	3	3
20516	20516	2017	3	4
20517	20517	2017	3	5
20518	20518	2017	3	6
20519	20519	2017	3	7
20520	20520	2017	3	8
20521	20521	2017	3	9
20522	20522	2017	3	10
20523	20523	2017	3	11
20524	20524	2017	3	12
20525	20525	2017	3	13
20526	20526	2017	3	14
20527	20527	2017	3	15
20528	20528	2017	3	16
20529	20529	2017	3	17
20530	20530	2017	3	18
20531	20531	2017	3	19
20532	20532	2017	3	20
20533	20533	2017	3	21
20534	20534	2017	3	22
20535	20535	2017	3	23
20536	20536	2017	3	24
20537	20537	2017	3	25
20538	20538	2017	3	26
20539	20539	2017	3	27
20540	20540	2017	3	28
20541	20541	2017	3	29
20542	20542	2017	3	30
20543	20543	2017	3	31

20544 rows × 4 columns

In [42]:

```
print(b_dates_)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

[illegible]

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 84

355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,  
13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 3  
7, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61  
, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,  
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 10  
8, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,  
128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147  
, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 1  
67, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186,  
187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206  
, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 2  
26, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245,  
246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265  
, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 2  
85, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304,  
305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324  
, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 3  
44, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363,  
364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 2  
5, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49  
, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73,  
74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 9  
8, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 1  
18, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137,  
138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157  
, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 1  
77, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196,  
197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216  
, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 2  
36, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255,  
256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275  
, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 2  
95, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314,  
315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334  
, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 3  
54, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,  
13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 3  
7, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61  
, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,  
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 10  
8, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,  
128, 129, 1

166, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165,  
166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185  
, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 2  
05, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224,  
225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244  
, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 2  
64, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283,  
284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303  
, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 3  
23, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342,  
343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362  
, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23  
, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,  
48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 7  
2, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96  
, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,  
117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136  
, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 1  
56, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175,  
176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195  
, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 2  
15, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,  
235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254  
, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 2  
74, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293,  
294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313  
, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 3  
33, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352,  
353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11  
, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,  
36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 6  
0, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84  
, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106,  
107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126  
, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 1  
46, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165,  
166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185  
, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 2  
05, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224,  
225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244  
, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 2  
6

[illegible]

, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94  
, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 1  
15, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134,  
135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154  
, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 1  
74, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193,  
194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213  
, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 2  
33, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252,  
253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272  
, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 2  
92, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311,  
312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331  
, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 3  
51, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8,  
9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33  
, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57,  
58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 8  
2, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 1  
05, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124,  
125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144  
, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 1  
64, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183,  
184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203  
, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 2  
23, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242,  
243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262  
, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 2  
82, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301,  
302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321  
, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 3  
41, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360,  
361, 362, 363, 364, 365, 366, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2  
0, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44  
, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,  
69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 9  
3, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 1  
14, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133,  
134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153  
, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 1  
73, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 1



213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 2  
32, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251,  
252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271  
, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 2  
91, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310,  
311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330  
, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 3  
50, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7  
, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 3  
2, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56  
, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,  
81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 1  
04, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123,  
124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143  
, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 1  
63, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,  
183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202  
, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 2  
22, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241,  
242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261  
, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 2  
81, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300,  
301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320  
, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 3  
40, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359,  
360, 361, 362, 363, 364, 365, 366, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 4  
3, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67  
, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,  
92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 1  
13, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132,  
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152  
, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 1  
72, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191,  
192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211  
, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 2  
31, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250,  
251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270  
, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 2  
90, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309,  
310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329  
, 33

[illegible]

142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 1  
61, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180,  
181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200  
, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 2  
20, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239,  
240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259  
, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 2  
79, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298,  
299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318  
, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 3  
38, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357,  
358, 359, 360, 361, 362, 363, 364, 365, 366, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16  
, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,  
41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 6  
5, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89  
, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 1  
11, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,  
131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150  
, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 1  
70, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189,  
190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209  
, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 2  
29, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248,  
249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268  
, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 2  
88, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307,  
308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327  
, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 3  
47, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2  
, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,  
28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 5  
2, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76  
, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100  
, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 1  
20, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139,  
140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159  
, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 1  
79, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198,  
199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218  
, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 2  
38, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257,  
258, 2

276, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296  
 , 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 3  
 16, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335,  
 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355  
 , 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13  
 , 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,  
 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 6  
 2, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86  
 , 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108,  
 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128  
 , 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 1  
 48, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167,  
 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187  
 , 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 2  
 07, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226,  
 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246  
 , 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 2  
 66, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285,  
 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305  
 , 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 3  
 25, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344,  
 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364  
 , 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,  
 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5  
 0, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74  
 , 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  
 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118,  
 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138  
 , 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 1  
 58, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177,  
 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197  
 , 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 2  
 17, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236,  
 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256  
 , 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 2  
 76, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295,  
 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315  
 , 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 3  
 35, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354,  
 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,

62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,  
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 10  
8, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,  
128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147  
, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 1  
67, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186,  
187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206  
, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 2  
26, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245,  
246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265  
, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 2  
85, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304,  
305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324  
, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 3  
44, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363,  
364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 2  
5, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49  
, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73,  
74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 9  
8, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 1  
18, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137,  
138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157  
, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 1  
77, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196,  
197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216  
, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 2  
36, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255,  
256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275  
, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 2  
95, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314,  
315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334  
, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 3  
54, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,  
13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 3  
7, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61  
, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,  
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 10  
8, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,  
128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147  
, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 1  
67, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186,  
187, 1

50, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 3

In [43]:

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

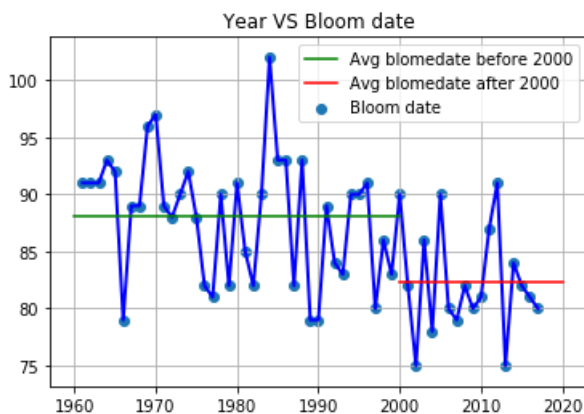
[illegible]

[illegible]

In [46]:

```
years_ = []
b_dates_ = []
for index, row in bloomDateDf.iterrows():
    year = int(row["year"])
    years_.append(year)
    month = int(row["month"])
    date = int(row["day"])
    nth_day_in_the_year = (dt.date(year, month, date) - dt.date(year, 1, 1)).days + 1
    b_dates_.append(nth_day_in_the_year)
index_of_2000 = years_.index(2000)

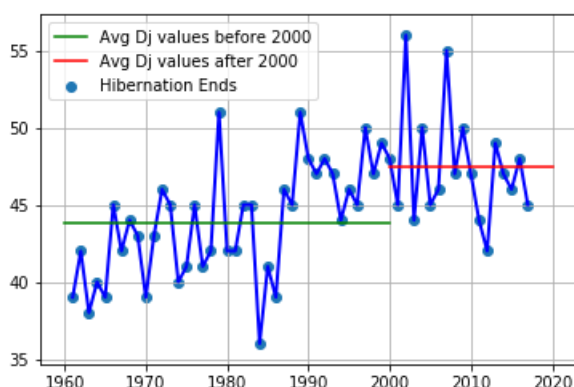
plt.plot(years_, b_dates_, color='blue', linewidth=2)
plt.scatter(years_, b_dates_, label="Bloom date")
plt.plot([x for x in range(1960, 2001)], [np.average(b_dates_[:index_of_2000]) for y in range(41)], color='green', label="Avg blomedate before 2000")
plt.plot([x for x in range(2000, 2021)], [np.average(b_dates_[index_of_2000:]) for y in range(21)], color='r', label="Avg blomedate after 2000")
plt.title('Year VS Bloom date')
plt.grid(True)
plt.legend()
plt.show()
```



**From the above graph we can see that,sakura started to bloom earlier from the previous years.**

In [50]:

```
plt.plot(years_,Dj,color='blue', linewidth=2)
plt.scatter(years_,Dj,label="Hibernation Ends")
plt.plot([x for x in range(1960,2001)], [np.average(Dj[:index_of_2000]) for y in range(41)],color='green',label="Avg Dj values before 2000")
plt.plot([x for x in range(2000,2021)], [np.average(Dj[index_of_2000:]) for y in range(21)],color='r',label="Avg Dj values after 2000")
plt.grid(True)
plt.legend()
plt.show()
```





sakura need more days to woke up from hibernation from the previous years so we can see sakura have a large hibernation time then before but blooms earlier.  
so it is clear that sakura adapeded itself to bloom in less temperature then before.