**Assign 2P user Requirements:**

The user must include a date(yyyy-mm-dd) as one of the arguments when running the class object

Then they must select either tomorrow or yesterday or the add/subtract methods which will use their date and a counter number to either add or subtract automatically based on the value they give ( - = subtract, + = add) Then they can also call the days\_to\_time method which will calculate the amount of days from the date of epoch(Jan 1 1970) then output a result which is an integer from the epoch date to their date. Then they can also call the day\_of\_week method which will give them the day of the week of their date in a 0-6 format. Ouputs only 0-6 based on the day.

**Algorithm**

Import os

Import sys

Create a class object = date

class Date:

def \_\_init\_\_(self, year, month, day):

self.year = year

self.month = month

self.day = day

Init function – Create the self.year/month/day object

def \_\_repr\_\_(self):

return '%.4d-%.2d-%.2d' % (self.year, self.month, self.day)

Repr – accepts the self object, formats & Returns the date in a yyyy-mm-dd format

#return date object as a string in "yyyy-mm-dd" format

def \_\_str\_\_(self):

return '%.4d/%.2d/%.2d' % (self.year, self.month, self.day)

Str – accepts the self object & formats & returns the date in a yyyy/mm/dd format

#return date object as a string in "yyyy/mm/dd" format

def \_\_add\_\_(self, other):

Addition overloading – accepts a date & a counter, in a while loops calculates the correct date for the counter which was inputted from the date which was entered

def \_\_sub\_\_(self, other):

Subtraction overloading – subtracts two dates from each other to find the difference between dates in a int value, returns the int inbetween the days/months/years

def tomorrow(self):

Tomorrow – Accepts the self object, calculates the next day from the inputted date

print("Tomorrow is ")

def yesterday(self):

Yesterday – accepts the self object, calculates the previous day from the inputted date

print("Yesterday was ")

def day\_of\_week(self):

Day of week – Accepts the self object, calculates which day of the week the date is using the day\_of\_week method of the object, returns day of week in a 0-6 format using the datetime module

print("It is " + day of week)

def days\_to\_time():

#convert an integer which is n days from epoch (Jan 1, 1970) to a corresponding date object.