SE 5930
Machine Problem 6 (MP6)
Stacks & Queues with Player Class

30 points

We are learning about **Stacks** & **Queues**. Write a Python script that reads data in from the text file `'5930 – MP6 Data.txt'`. The data file contains information about players on a roster. They have a number and a name. Each player will be randomly pushed onto a Stack, or enqueued into a Queue. Then, a series of 20 random moves will take place. There are 4 possible moves…

(1) `S.pop()` then `Q.enqueue()`
(2) `S.pop()`
(3) `Q.dequeue()` then `S.push()`
(4) `Q.dequeue()`.

The results of each move will be reported, along with the status of the Queue and the Stack after each move. Your script must be able to handle the possibility that the Stack or Queue could be empty at any time.

- First, define a `class` named `Player` with the details described below. Define the `class` in a separate file, and then `import` it at the beginning of your program.

```
#
# Player Class Definition
#
# 2 Instance Variables:
#       _number         int
#       _name           str - first & last name together in 1 str
#
# Constructor:
#       __init__(self, number, name)    Initialize _number to number,
#                                       _name to name
#
# 4 Public Instance Methods:
#       getNumber(self)         return _number
#       getName(self)           return _name
#       setNumber(self, number) set _number to number
#       setName(self, name)     set _name to name
#

class Player:
```

< The instance Variables and Instance Methods for `Player` are defined here >

- For our Stack implementation, we will use the `ArrayStack` definition in section 6.1 of our Data Structures text (I just copied and pasted into a separate file). For our Queue implementation, we will use the `ArrayQueue` definition in section 6.2. (There is 1 typo in the `ArrayQueue` definition… there is a `.data` that should be `._data`). Put both of these definitions in a separate file and then `import` them at the beginning of your program.

- Since there are 2 `random` actions being taken in the problem, you will also need to import the `random` class. There are a number of ways to get the `random` actions executed… I used `choice()` for both.

- Your main program will…

  (1) instantiate (create) a Stack and a Queue
  (2) call a method named `getPlayers()` that reads the information in from the data file, putting each player's information randomly onto the Stack or into the Queue
  (3) print the Initial Status of the Stack and the Queue
  (4) execute the 20 random moves

- Your program will use 2 methods… `getPlayers()` and `currentStatus()`. The documentation for both is given below…

```
def getPlayers(myS, myQ):

    #
    # Opens the data file of player info... number, firstName, lastName... reads
    # each line of data as a str, divides the line into the 3 values...
    # str, str, str... converts the 1st str into an int (number)... combines the
    # 2nd & 3rd str objects into 1 str (name)... instantiates a new Player object,
    # using the constructor to set the number & name values... then, randomly
    # pushes the object onto a stack, or enqueues the object into a queue...
    #
    #
    # myS       An object of the ArrayStack class
    #           Initially empty - Player objects randomly pushed onto myS
    #
    # myQ       An object of the ArrayQueue class
    #           Intitially empty - Player objects randomly enqueued into myQ
    #
    # No return value
    #
```

< the body of your function `getPlayers()` goes here >

```
def currentStatus(myS, myQ):

    #
    # Prints the contents of the Player object at the top of the stack, along with the
    # number of elements currently in the stack.  If the stack is empty, an appropriate
    # message is indicated.  Then, does the same for the Player object at the front of
    # the queue.  If the queue is empty, an appropriate message is indicated.
    #
    # myS       An object of the ArrayStack class - could be empty
    #
    # myQ       An object of the ArrayQueue class - could be empty
    #
    # No return value
    #
```

< the body of your function `currentStatus()` **goes here >**

Your output must be formatted like this…

```
Initial Status:
Stack Top: 6 Lynn Williams(9)... Queue Front: 18 Casey Murphy (7)

Move 1:
Pop: 6 Lynn Williams.
Stack Top: 20 Trinity Rodman(8)... Queue Front: 18 Casey Murphy (7)

Move 2:
Dequeue: 18 Casey Murphy then Push.
Stack Top: 18 Casey Murphy(9)... Queue Front: 19 Crystal Dunn (6)

Move 3:
Dequeue: 19 Crystal Dunn then Push.
Stack Top: 19 Crystal Dunn(10)... Queue Front: 5 Kelley OHara (5)

Move 4:
Dequeue: 5 Kelley OHara then Push.
Stack Top: 5 Kelley OHara(11)... Queue Front: 8 Julie Ertz (4)

Move 5:
Dequeue: 8 Julie Ertz.
Stack Top: 5 Kelley OHara(11)... Queue Front: 16 Rose Lavelle (3)

Move 6:
Pop: 5 Kelley OHara.
Stack Top: 19 Crystal Dunn(10)... Queue Front: 16 Rose Lavelle (3)

Move 7:
Pop: 19 Crystal Dunn.
Stack Top: 18 Casey Murphy(9)... Queue Front: 16 Rose Lavelle (3)

Move 8:
Dequeue: 16 Rose Lavelle then Push.
Stack Top: 16 Rose Lavelle(10)... Queue Front: 17 Andi Sullivan (2)

Move 9:
Dequeue: 17 Andi Sullivan then Push.
Stack Top: 17 Andi Sullivan(11)... Queue Front: 11 Sophia Smith (1)

Move 10:
Dequeue: 11 Sophia Smith then Push.
Stack Top: 11 Sophia Smith(12)... Queue is empty (0).

Move 11:
Dequeue: Queue is empty.
Stack Top: 11 Sophia Smith(12)... Queue is empty (0).

Move 12:
```

```
Dequeue: Queue is empty.
Stack Top: 11 Sophia Smith(12)... Queue is empty (0).

Move 13:
Dequeue: Queue is empty.
Stack Top: 11 Sophia Smith(12)... Queue is empty (0).

Move 14:
Pop: 11 Sophia Smith then Enqueue.
Stack Top: 17 Andi Sullivan(11)... Queue Front: 11 Sophia Smith (1)

Move 15:
Pop: 17 Andi Sullivan.
Stack Top: 16 Rose Lavelle(10)... Queue Front: 11 Sophia Smith (1)

Move 16:
Dequeue: 11 Sophia Smith then Push.
Stack Top: 11 Sophia Smith(11)... Queue is empty (0).

Move 17:
Dequeue: Queue is empty.
Stack Top: 11 Sophia Smith(11)... Queue is empty (0).

Move 18:
Pop: 11 Sophia Smith then Enqueue.
Stack Top: 16 Rose Lavelle(10)... Queue Front: 11 Sophia Smith (1)

Move 19:
Pop: 16 Rose Lavelle then Enqueue.
Stack Top: 18 Casey Murphy(9)... Queue Front: 11 Sophia Smith (2)

Move 20:
Pop: 18 Casey Murphy.
Stack Top: 20 Trinity Rodman(8)... Queue Front: 11 Sophia Smith (2)
```

Notice that the Queue is empty in this example. Here is another example with the Stack being empty…

```
Initial Status:
Stack Top: 2 Ashley Sanchez(4)... Queue Front: 21 Aubrey Kingsbury (12)

Move 1:
Pop: 2 Ashley Sanchez then Enqueue.
Stack Top: 16 Rose Lavelle(3)... Queue Front: 21 Aubrey Kingsbury (13)

Move 2:
Pop: 16 Rose Lavelle then Enqueue.
Stack Top: 10 Lindsey Horan(2)... Queue Front: 21 Aubrey Kingsbury (14)

Move 3:
Pop: 10 Lindsey Horan.
Stack Top: 8 Julie Ertz(1)... Queue Front: 21 Aubrey Kingsbury (14)

Move 4:
Dequeue: 21 Aubrey Kingsbury then Push.
Stack Top: 21 Aubrey Kingsbury(2)... Queue Front: 18 Casey Murphy (13)

Move 5:
Pop: 21 Aubrey Kingsbury then Enqueue.
Stack Top: 8 Julie Ertz(1)... Queue Front: 18 Casey Murphy (14)

Move 6:
Pop: 8 Julie Ertz.
Stack is empty (0)... Queue Front: 18 Casey Murphy (14)

Move 7:
```

Dequeue: 18 Casey Murphy then Push.
Stack Top: 18 Casey Murphy(1)... Queue Front: 12 Alana Cook (13)

Move 8:
Pop: 18 Casey Murphy then Enqueue.
Stack is empty (0)... Queue Front: 12 Alana Cook (14)

Move 9:
Pop: Stack is empty.
Stack is empty (0)... Queue Front: 12 Alana Cook (14)

Move 10:
Dequeue: 12 Alana Cook then Push.
Stack Top: 12 Alana Cook(1)... Queue Front: 19 Crystal Dunn (13)

Move 11:
Pop: 12 Alana Cook.
Stack is empty (0)... Queue Front: 19 Crystal Dunn (13)

Move 12:
Pop: Stack is empty.
Stack is empty (0)... Queue Front: 19 Crystal Dunn (13)

Move 13:
Dequeue: 19 Crystal Dunn then Push.
Stack Top: 19 Crystal Dunn(1)... Queue Front: 3 Sofia Huerta (12)

Move 14:
Pop: 19 Crystal Dunn.
Stack is empty (0)... Queue Front: 3 Sofia Huerta (12)

Move 15:
Dequeue: 3 Sofia Huerta then Push.
Stack Top: 3 Sofia Huerta(1)... Queue Front: 5 Kelley OHara (11)

Move 16:
Pop: 3 Sofia Huerta.
Stack is empty (0)... Queue Front: 5 Kelley OHara (11)

Move 17:
Pop: Stack is empty.
Stack is empty (0)... Queue Front: 5 Kelley OHara (11)

Move 18:
Dequeue: 5 Kelley OHara then Push.
Stack Top: 5 Kelley OHara(1)... Queue Front: 22 Kristie Mewis (10)

Move 19:
Pop: 5 Kelley OHara.
Stack is empty (0)... Queue Front: 22 Kristie Mewis (10)

Move 20:
Pop: Stack is empty.
Stack is empty (0)... Queue Front: 22 Kristie Mewis (10)

For full credit…

- Hand in a hard copy of your Player class definition (with documentation), your main program script (with documentation), and the output that your script generates.
- Hand in your solution on-time.
- Document your solution appropriately.
- Write clear, well-organized code.
- Use variable names that make sense for the data that they contain.
- Please, hand in 2 sample runs, one where the Stack is empty, and one where the Queue is empty, so that I can verify that your program handles both.