MR2020: Extra Practice Problems

1. Given an array called values:

   ```
   values = np.array([5.7, 1.2, 9.3, 4.8, 7.6, 2.1, 0.9, 3.4, 6.5, 8.9])
   ```

   Write a for-loop or list comprehension to create a variable ‘count’ that stores the number of elements in ‘values’ that are greater than 5.

```
# For loop
count = 0
for k in values:
    if k > 5:
        count += 1

# List comprehension
len([i for i in values if i > 5])
```

2. What is the value of B after the following code is executed?

```
B = 3
for i in range(4):
    B += 3
```

15

3. Create a list called ‘num_list’ that contains the numbers 1, 4, and 7.

```
num_list = [1,4,7]
```

4. Create a tuple called num_tuple that contains the numbers 10, 20, and 30.

```
num_tuple = (10,20,30)
```

5. Create a dictionary called location_data containing the following information:

Key	Value
State	California
Elevation	300

Population	500000

```
location_data = {'State':'California','Elevation':300,'Population':500000}
```

6. Write Python statements equivalent to the following English statements:

a. X is equal to Y.

```
X == Y
```

b. Z is less than or equal to the median of a numpy array arr.

```
Z <= np.median(arr)
# or
Z <= arr.median()
```

c. Y is greater than the minimum value in arr.

```
Y > arr.min()
# or
Y > np.min(arr)
```

d. X is greater than Y and Y is greater than Z.

```
X > Y and Y > Z
# or
X > Y > Z
```

e. Either both X is equal to Y and X is less than Z or X is greater than Z.

```
(X == Y and X < Z) or X > Z
```

7. Translate the following Python statements into written English:

   a. (A + B) >= 5 and C < 2

   The sum of A and B are greater than or equal to 5 and C is less than 2.

   b. X != Y – 2

   X is not equal to Y minus 2.

   c. if Z > 10: Z -= 3

   If Z is greater than 10, reassign Z itself minus 3.

   d. A > B or A == 5

   A is greater than B or A is equal to 5.

   e. np.median(X[:, 1]) <= Y

   The median of the second column of X is less than or equal to Y.

8. Write a code snippet that attempts to create an array B filled with ones using np.ones_like(A) where A is another numpy array. Print "Assignment failed" if the operation fails.

```
try:
     arr = np.ones_like(A)
except:
     print('Assignment failed')
```

9. Write a while loop that generates a random integer from 1 to 20 using random.randint(1, 20) and continues until the generated number is 15.

```
number = 0  # Initialize the variable to a value that isn't
15
while number != 15:
    number = random.randint(1, 20)
    print(number)
```

10. Suppose you have arrays of data for pressure (P) and wind speed (WS):

Write a function check_conditions that returns 1 if the pressure is below 1000 hPa and wind speed exceeds 20 m/s. Otherwise, return 0.

```
def check_conditions(P, WS):
    if P < 1000 and WS > 20:
        return 1
    else:
        return 0
```

11. Define a class called Rectangle that is initialized with the attributes length (float), width (float), and color (string). Add a method to the class that calculates the area of the rectangle. Then, create an object my_rectangle with length 15, width 10, and color "blue" in one line.

```
class Rectangle:
    def __init__(self, length, width, color):
        self.length = length
        self.width = width
        self.color = color

    def area(self):
        return self.length * self.width

my_rectangle = Rectangle(length=15, width=10, color="blue")
```

12. For each problem below, determine the output of the Python code snippet:

a.

```
B = np.array([(1,4),(2,8)])
B[1, 0]
```

```
2
```

b.

```
C = np.array([5,9,2,6,1,8,3,4,7])
C[5:]
```

```
array([8,3,4,7])
```

c.

```
[k for k in range(6)]
```

```
[0,1,2,3,4,5]
```

d.

```
weather_info = {'temp': [20, 25, 30], 'wind': ['Yes', 'No', 'Yes']}
weather_info['wind'][2]
```

```
‘Yes’
```

13. Create a list called heights that contains the numbers 150, 165.2, 170, and 180.5 in that order.

```
heights = [150,165.2,170,180]
```

14. Create a tuple called weights that contains the numbers 60, 75.5, 82, and 70.3 in that order.

```
weights = (60,75.5,82,70.3)
```

15. Create a dictionary called book_info containing the following information:

Key	Value
Title	List containing elements ‘1984’ and ‘Brave New World’
Author	List containing elements ‘Orwell’ and ‘Huxley’

```
bookinfo = {'Title':['1984','Brave New World'],'Author':['Orwell','Huxley']}
```

16. Given the following Python set and the print statement:

```
fruits = {'apple', 'banana', 'apple', 'orange', 'banana'}
print(fruits)
```

What would you expect to see as the result of print(fruits)?

```
{'orange', 'apple', 'banana'}
```

17. What would the following code return for the variable B?

```
B = 1
for i in range(3):
    B += 2
    B *= 2
```

36

18. Write a Python comparison statement for each of the following English statements. Assume that numpy was imported as np.

a. The absolute difference between X and Y is less than or equal to 5. Hint: Use np.abs()

```
np.abs(X-Y) <= 5
```

b. M is greater than or equal to the square root of N. Use np.sqrt(N).

```
M >= np.sqrt(N)
```

c. Z is positive, or both P and Q are greater than or equal to 20.

```
Z > 0 or (P >= 20 and Q >= 20)
```

19. Translate the following Python statements to written English:

a. `A % B == 0 and B > A // 2`

The remainder when B is divided into A is 0 and B is greater than the floor of A divided by 2.

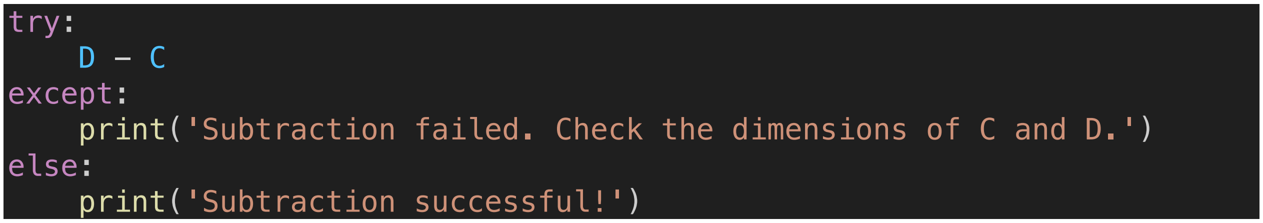
b. `C.min() < D[:,0].min()`

The minimum of C is less than the minimum of the first column of D.

c. `if not A: C -= 3`

If A is False, 0, or empty, then C is reassigned itself minus 3.

20. Write a Python code that tries to subtract C from D (i.e., D - C). If the subtraction fails, print "Subtraction failed. Check the dimensions of C and D." If the subtraction succeeds, print "Subtraction successful!"



```
try:
    D - C
except:
    print('Subtraction failed. Check the dimensions of C and D.')
else:
    print('Subtraction successful!')
```

21. Assume you have the following code:

```
from random import randint
```

Use the following line of code inside a while loop

```
num = randint(1, 4)
```

If num is not 4, print the number. If num is 4, set a variable called found to True and exit the while loop.

```
from random import randint

found = False  # Initialize the variable to control the loop

while not found:
    num = randint(1, 4)  # Generate random integer from 1 to 4
    if num != 4:
        print(num)  # Print the number if it is not 4
    else:
        found = True  # Set found to True and exit
```

22. Suppose you already have the following code:

```
import numpy as np

X = np.array([1.2, 3.5, 6.7, 2.1, 8.4])
Y = np.array([4.3, 1.5, 2.9, 5.6, 3.8])
```

Write a code that prints how many times the sum of corresponding elements in X and Y is less than or equal to 10.

```
(X + Y <= 10).sum()
```

or

```
np.sum(X + Y <= 10)
```

23. Define a class called WeatherStation that is initialized with the following attributes: temperature (float), pressure (float), humidity (float), and wind_speed (float). Add a method that calculates the wind chill using the formula:

$$WC = 13.12 + 0.6215 \times T - 11.37 \times V^{0.16} + 0.3965 \times T \times V^{0.16}$$

where T is temperature and V is wind speed. Create an object station with temperature of 5, pressure of 1013, humidity of 85, and wind_speed of 10, then calculate the wind chill using this object. What answer do you get for the wind chill?

```
class WeatherStation:
    def __init__(self, temperature, pressure, humidity, wind_speed):
        self.temperature = temperature
        self.pressure = pressure
        self.humidity = humidity
        self.wind_speed = wind_speed

    def calculate_wind_chill(self):
        T = self.temperature
        V = self.wind_speed
        # Wind chill formula
        WC = 13.12 + 0.6215 * T - 11.37 * (V ** 0.16) + 0.3965 * T * (V ** 0.16)
        return WC

station = WeatherStation(5,1013,85,10)

\
```

2.66

Write a function called calculate_heat_index that takes two input variables: T (temperature in degrees Fahrenheit) and RH (relative humidity as a percentage). The function should return the Heat Index (HI) using the following made-up formula:

$$HI = -42.379 + 2.049 \times T + 10.143 \times RH - 0.225 \times T \times RH$$

The function should warn by printing a message if any special cases where the input values lead to extreme or unrealistic Heat Index values (for example greater than 135 degrees F).

```python
def calculate_heat_index(T, RH):
    HI = -42.379 + 2.049*T + 10.143*RH - 0.225*T*RH
    if HI > 135:
        Warning('The heat index is very high.')
    return HI
```

25. For each separate, unrelated problem, determine what the following Python code snippets would return:

a.

```
B = np.array([(6,2,3,9),(7,5,1,4),(8,0,2,6)])
B[1, 2]
```

1

b.

```
B = np.array([2,4,6,8,10,12,14,16,18,20])
B[2:]
```

```
array([ 6,  8, 10, 12, 14, 16, 18, 20])
```

c.

```
B = np.array([2,4,6,8,10,12,14,16,18,20])
B[1:7]
```

d.

```
B = np.array([2,4,6,8,10,12,14,16,18,20])
B[::4]
```

e.

```
data = {'season':[1,2,'Offseason',4],'played':[False,True,True,False]}
data[season][2]
```