

## MR2020 (Autumn 2021) Midterm Practice/Review

1. Suppose a bag contains 3 red balls, 2 blue balls, and 1 white ball. You want to estimate how many times you would pick each color ball if you pulled one ball from the bag at a time then immediately put it back and repeated that action 100,000 times. Write a python code that accomplishes this using a for loop. At the end, print the number of each color ball.

```
import numpy as np
red, blue, white = 0, 0, 0
for i in range(100000):
    I = np.random.randint(6)
    if I <= 2: # red
        red += 1
    elif I == 5: # white
        white += 1
    else: # blue
        blue += 1
print(str(red) + ' red balls')
print(str(blue) + ' blue balls')
print(str(white) + ' white balls')
```

2. What would the following return for the variable A?

```
>> A = 2
for i in range(5):
    A *= 2
```

64

3. List each python operator and explain what it does.

- + Addition
- Subtraction
- \* Multiplication
- / Division
- \*\* Exponent
- // Floor divider (returns the next integer lower than the quotient of two numbers)
- % Modulus (returns the remainder in the quotient of two numbers and is the same sign as the divisor)

4. Write a Python comparison statement for each of the following English statements. You can assume that numpy was imported as np.

a. A is not equal to B.

```
A != B
```

b. X is greater than or equal to the mean of numpy array B.

```
X >= np.mean(B) Other possible answer: X >= B.mean()
```

c. X is equal to the maximum value in the second row of two-dimensional numpy array B.

```
X == np.max(B[2,:])
```

d. A is less than B and B is less than C.

```
A < B < C Other possible answer: A < B and B < C
```

e. Either both A is equal to B and A is less than C or just A is greater than C.

```
(A == B and A < C) or (A > C)
```

5. Write a sample code that attempts to assign the variable A an array of zeros with the same dimensions of B; however, prints "This operation failed" if the variable assignment fails. Assume that numpy is already imported as np.

```
try:
    A = np.zeros_like(B)
except:
    print('This operation failed.')
```

6. Generate a while loop that creates a random integer from 0 to 10, counts the number of times that the integer is 6, and stops only when the number of times the integer was 6 reaches 1000.

```
import numpy as np

ct = 0
while ct < 1000:
    num = np.random.randint(11)
    if num == 6:
        ct += 1
```

7. Suppose you have the following arrays of data for temperature (T) and relative humidity (RH) and that numpy is imported as np.

```
>> T = 30*np.random.random(1000)+5    # degrees C
>> RH = 100*np.random.random(1000)    # %
```

Write a function that returns 1 if the temperature exceeds 20°C and RH exceeds 80% but returns 0 if that condition is not met.

Then, inside a for-loop that loops through each entry in the arrays T and RH, pass the values of T and RH to the function individually and at the conclusion of the for-loop, print the number of times that the above condition is met.

```
def myfunc(T,RH):
    if T > 20 and RH > 80:
        return 1
    else:
        return 0
```

```
ct = 0
for i in range(T.size):
    ct += myfunc(T[i],RH[i])

print('Condition was met ' + str(ct) + ' times.')
```

\*\*\*Note: There is a faster way to do this without a for-loop, but this isn't part of the answer to this question. You could do this:

```
ct = T[(T>20)*(RH>80)].size
print('Condition was met ' + str(ct) + ' times.')
```

8. Define a class called Cloud that must be initialized with the following attributes (with data type for each attribute in parentheses): base (float), height (float), width (float), raining (boolean). Give the class a method (function) that computes and prints the depth (height minus base) of the cloud.

```
class Cloud:
    def __init__(self,base,height,width,raining):
        self.base = base
        self.height = height
        self.width = width
        self.raining = raining

    def depth(self):
        return self.height-self.base

cloud = Cloud(.5,13,4,True)
print('Cloud depth is ' + str(cloud.depth()) + '.')
```

9. Translate the following Python statements to written English.

a.  $(A+B) < 2$  and  $A \geq 5$

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

b.  $A \neq B + 1$

A is not equal to B plus 1.

c. if  $B < 5$ :  $B += 1$

If B is less than 5, then add 1 to B.

d.  $A \leq B$  or  $A == 0$

A is less than or equal to B or A is equal to zero.

e.  $\text{np.mean}(A[:,0]) > B$

The numerical mean of A along its 0<sup>th</sup> column is greater than B.

10. For each separate, not related, problem, respond with what the following **bolded red** lines of code would return (or be evaluated as) in Python.

a. **[j for j in range(5)]**

**[0, 1, 2, 3, 4]**

b.

```
A = np.array([(2,3),(4,5)])
```

**A[0,1]**

**3**

c.

```
def C2F(tempC):  
    import numpy as np  
  
    return tempC*1.8 + 32
```

**C2F(30)**

**86**

d.

```
class Temperature:  
  
    def __init__(self, temp, units):  
        self.T = temp  
        self.units = units  
  
    def TtoC(self):  
        if self.units == 'degF':  
            return (self.T-32)/1.8  
  
        elif self.units == 'degC':  
            return self.T  
  
        elif self.units == 'K':  
            return self.T - 273.15
```

```
T = Temperature(50, 'degF')
```

**T.TtoC()**

**10**

e.

```
weather =  
{ 'temp': [43, 45, 47], 'rain': [ 'True', 'True', 'False' ] }  
weather[ 'temp' ] [ 2 ]
```

47

11. Write UNIX statements that do the following from a UNIX terminal.

a. List all nonhidden files and directories at the path `/home/MR2020/code`

```
ls /home/MR2020/code
```

b. Show the path to the current directory.

```
pwd
```

c. Change a file called `data.dat` in your current path to a file called `data.txt`.

```
mv data.dat data.txt
```

d. Delete the file `olddata.nc4`.

```
rm olddata.nc4
```

e. Go to your home directory.

```
cd
```

f. Go to the directory `/home/MR2020/code`.

```
cd /home/MR2020/code
```

12. Will the following lines or blocks of code cause an error? If so, why?

a. `[j for j in range(100):]`

Error: Colon should not appear after for statement in list comprehension.

b.

```
A = 0
while A > 1:
    A -= 1
```

No error. In fact, the while loop never executes.

c.

```
def pythag(a,b):
    import numpy as np
    return np.sqrt(a**2+b**2)

C = pythag(2,3,4)
```

Error: The function pythag requires 2 input arguments, but 3 arguments are passed.

d.

```
def pythag(a,b):
    import numpy as np
    return np.sqrt(x**2+b**2)

C = pythag(2,3)
```

Error: Reference to undefined variable 'x' in function pythag. (Variables should be called in function as a or b because that is what they are named in the function declaration.)

e.

```
people = {'name':['sally','jake'],'age':[42,35]}
```

No error.

13. Suppose a few lists are defined as below.

```
temperature = [30,31,29,27,28]
salinity = [24,25,24,23,24]
wind = [10,8,5,14,4]
```

Using these lists, use one line of code to create a dictionary with variable name *ocean* that uses the following keys (*T*, *S*, and *W*) paired respectively with the lists *temperature*, *salinity*, and *wind* as corresponding values.

```
ocean = dict{'T':temperature,'S':salinity,'W':wind}
```

14.

a. Write down the line of code that would recast an array *A* as a list.

```
A = list(A)
```

b. Write down the line of code that would recast a list *A* as a numpy array.

```
A = np.array(A)
```