Test 1

August 18, 2022

Name: Amartya Sinha

Roll No: AC-1207

Data Analysis & Visualization

[ ]:
```
############################################################
```

Q1. Given a list of names, use list comprehension to reverse those names which starts with letter 'Z' and ends with letter 'a' (4)

[ ]:
```
names = ['abc', 'xyz', 'zoa']
print("Original Names: ", names)

new_names = [name[::-1] if name.startswith('z') and name.endswith('a') else␣
 ↪name for name in names]
# for name in names:
#     if name.startswith('z') and name.endswith('a'):
#         name = name[::-1]

print("Names After Reverse on Condition: ", new_names)
```

```
Original Names:  ['abc', 'xyz', 'zoa']
Names After Reverse on Condition:  ['abc', 'xyz', 'aoz']
```

Q2 Create an array for marks in 5 subjects for 6 students where marks are in (0-100). Take another array having names of 6 students. Do the following: (3+3+3)

a. Find name of students with average marks > 90

b. Count number of students having marks more than that of the average of class

c. For each student display the number of subjects in which they have scored > 60 marks

and display the following:

I. Name of students number of subjects where marks>60

[ ]:
```
import numpy as np

marks = np.random.randint(0,100,(5,6))
```

```
print("Marks of 6 Students in 5 Subjects:\n", marks)

names = ['a', 'b', 'c', 'd', 'e', 'f']
print("Names of 6 Students", names)
```

```
Marks of 6 Students in 5 Subjects:
 [[22 17  4 55 60 27]
 [ 9 29 39 60 67 84]
 [52 71 92 29 64 17]
 [17 63  3 68 75 52]
 [27 21 51 98 50 35]]
Names of 6 Students ['a', 'b', 'c', 'd', 'e', 'f']
```

[ ]:
```
avg_marks = np.average(marks, axis=0)
print("Avg Marks of Students: ", avg_marks)
ind = np.where(avg_marks>60)[0]    # since no student has avg > 90

for i in ind:
    print(names[i])
```

```
Avg Marks of Students:  [25.4 40.2 37.8 62.  63.2 43. ]
d
e
```

[ ]:
```
class_avg = np.average(marks)
print("Class Avg Marks: ", class_avg)

print("No of Students with more than Class Avg: ", np.
  ↪count_nonzero(avg_marks>class_avg))
```

```
Class Avg Marks:  45.266666666666666
No of Students with more than Class Avg:  2
```

[ ]:
```
print("No of Students where marks>60 for each Student")
for (i, sub_marks), each_stu_marks in zip(enumerate(marks), marks.T):
    print(names[i], end=": ")
    print(np.count_nonzero(each_stu_marks>60))
```

```
No of Students where marks>60 for each Student
a: 0
b: 2
c: 1
d: 2
e: 3
```

Q3. Create a 3x4 array of random numbers. Write two ways to set its second row to zero and third row to maximum value in the original matrix (2+2)

[ ]:
```
my_arr = np.random.randn(3,4)
print(my_arr)
```

```
[[-1.32378764 -0.35258972 -0.47116928  0.20772276]
 [ 0.97878502 -0.6991125  -0.40136923  0.42825752]
 [-1.24188061 -0.77870287  0.40086077  0.07631309]]
```

[ ]:
```
my_arr[[1]] = 0
my_arr[[2]] = np.amax(my_arr)

print(my_arr)
```

```
[[-1.32378764 -0.35258972 -0.47116928  0.20772276]
 [ 0.          0.          0.          0.        ]
 [ 0.40086077  0.40086077  0.40086077  0.40086077]]
```

[ ]:
```
another_arr = np.random.randn(3,4)
print(another_arr)
```

```
[[-0.65561574 -0.39418658 -2.02337197 -0.42476684]
 [ 0.09275159 -1.29651915 -0.7154588  -0.34614912]
 [-0.41921325  0.01697083 -1.70087202  0.02590159]]
```

[ ]:
```
another_arr[1:2] = 0
another_arr[2:] = np.amax(another_arr)

print(another_arr)
```

```
[[-0.65561574 -0.39418658 -2.02337197 -0.42476684]
 [ 0.          0.          0.          0.        ]
 [ 0.02590159  0.02590159  0.02590159  0.02590159]]
```

Q4. Write a lambda function which takes an element x, and returns the reverse of the element in uppercases. Make sure element passed must be of string type else raise exception. (3)

[ ]:
```
fn = lambda x: x[::-1].upper()

my_inp = input("Enter the String you want to reverse:")
try:
    if my_inp.isdigit():
        raise ValueError
    else:
        print(fn(my_inp))
except ValueError:
    print("Only Strings are allowed")
```

```
OLLEH
```