Dennis Johnson – 180905025
CSE – B1 – 9/3/2021

Lab3 – Python Practice 3

1.Write a program to find the factors of a given number (get input from user) using for loop.

```
num = int(input("Enter a number to factorize "))


factors = set()


for i in range(2, num):
        if num % i == 0:
                factors.add(i)


print('The factors of {} are {}'.format(num, factors))
```

```
dennis@project-lab:~/ds-lab/lab3$ python factor.py
Enter a number to factorize 32
The factors of 32 are set([8, 16, 2, 4])
```

2. Find the sum of columns and rows using axis.

```
import numpy as np

mat = np.ones((3,4))

print("Original Matrix is:")
print(mat)

print("Sum of rows, axis=0 ---> {}".format(np.sum(mat, axis=0)))
print("Sum of cols, axis=1 ---> {}".format(np.sum(mat, axis=1)))
```

```
dennis@project-lab:~/ds-lab/lab3$ python axis-sum.py
Original Matrix is:
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
Sum of rows, axis=0 ---> [3. 3. 3. 3.]
Sum of cols, axis=1 ---> [4. 4. 4.]
```

3. Operations on Arrays (use numpy wherever required):

a) Create array from list with type float

b) Create array from tuple

c) Creating a 3X4 array with all zeros

d) Create a sequence of integers from 0 to 20 with steps of 5

e) Reshape 3X4 array to 2X2X3 array

f) Find maximum and minimum element of array, Row wise max and min, column wise max and min and sum of elements. (Use functions max(), min(), sum())

```
import numpy as np

#-----------------------------------------------
print("a) Array with float32 data type:")
fl = np.ones((1,3), dtype = np.float32)
print(fl)

#-----------------------------------------------
tup = (11, 22, 23, 24, 55, 67)
print("\nb) Creating np array from the tuple")
print(tup)

tupArr = np.array(tup)
print(tupArr)

#-----------------------------------------------
print("\nc) Creating a 3x4 matrix of zeros")
mat3x4 = np.zeros((3,4))
print(mat3x4)

#-----------------------------------------------
print("\nd) Sequence of integers from 0-20 with step size 5")
seq = np.arange(0,20+1,5) #end is non-inclusive
print(seq)


#-----------------------------------------------
print("\ne) Reshape 3x4 matrix into a 2x2x3 matrix:")
reshapemat = np.random.rand(3,4)
print("Original matrix:", reshapemat)
reshapemat = np.reshape(reshapemat, (2,2,3))
print("Afer reshape: ", reshapemat)

#-----------------------------------------------
mat = np.arange(9).reshape(3,3)
print("\nf) Find min and max in the matrix: ", mat)
```

```
print("\nMax = {}, Min = {}".format(np.max(mat), np.min(mat)))
print("Row wise max = {}, min = {}".format(str(np.max(mat, axis=0)), np.min(mat, axis = 0)))
print("Col wise max = {}, min = {}".format(str(np.max(mat, axis=1)), np.min(mat, axis = 1)))
print("Sum of all elements = {}".format(np.sum(mat)))
```

```
a) Array with float32 data type:
[[1. 1. 1.]]

b) Creating np array from the tuple
(11, 22, 23, 24, 55, 67)
[11 22 23 24 55 67]

c) Creating a 3x4 matrix of zeros
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]

d) Sequence of integers from 0-20 with step size 5
[ 0  5 10 15 20]
```

```
d) Sequence of integers from 0-20 with step size 5
[ 0  5 10 15 20]

e) Reshape 3x4 matrix into a 2x2x3 matrix:
('Original matrix:', array([[0.90683324, 0.89666809, 0.2384692 , 0.39597507],
       [0.91830458, 0.38112152, 0.36542833, 0.27243587],
       [0.7748937 , 0.35192675, 0.81845355, 0.20725622]]))
('Afer reshape: ', array([[[0.90683324, 0.89666809, 0.2384692 ],
        [0.39597507, 0.91830458, 0.38112152]],

       [[0.36542833, 0.27243587, 0.7748937 ],
        [0.35192675, 0.81845355, 0.20725622]]]))
('Find min and max in the matrix: ', array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]]))

Max = 8, Min = 0
Row wise max = [6 7 8], min = [0 1 2]
Col wise max = [2 5 8], min = [0 3 6]
Sum of all elements = 36
```

4. **Write a program to transpose a given matrix.**

```
import numpy as np

mat = np.arange(9).reshape(3,3)
print("Transposing the given matrix", mat)

mat = mat.T
print("Transposed matrix is now: ", mat)
```

```
dennis@project-lab:~/ds-lab/lab3$ python transpose.py
('Transposing the given matrix', array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]]))
('Transposed matrix is now: ', array([[0, 3, 6],
       [1, 4, 7],
       [2, 5, 8]]))
```

5. **Write a program to add two matrices.**

```
import numpy as np

A = np.arange(6).reshape(3,2)
B = np.random.randn(3,2)

print("A:", A)
print("B:", B)

print("\nSum of A, B: ")
print(np.add(A, B))
```

```
dennis@project-lab:~/ds-lab/lab3$ python addMat.py
('A:', array([[0, 1],
       [2, 3],
       [4, 5]]))
('B:', array([[ 0.91501556,  0.87829979],
       [-0.86462605, -0.35389671],
       [ 0.87314964,  0.47709253]]))

Sum of A, B:
[[0.91501556 1.87829979]
 [1.13537395 2.64610329]
 [4.87314964 5.47709253]]
```

6. **Write a program to find element wise product between two matrices.**

```
import numpy as np

print("A:")
A = np.arange(6).reshape(2,3)
print(A)

print("B:")
B = np.arange(6).reshape(2,3) + 1
print(B)

print("Element wise product is:")
print(np.multiply(A, B))
```