

-: DS-LAB-WEEK : 3 :-

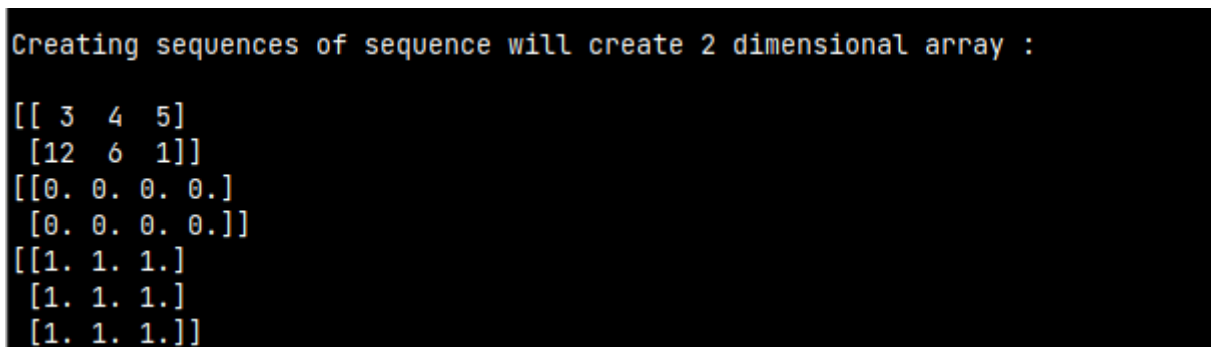
```
import numpy as np
import random
print("Array Creation : \n\n")
A = np.array([2, 5, 10])
print(A.dtype)
B = np.array([2.4, 10.6, 5.2])
print(B.dtype)
```

A terminal window titled "Terminal" with a dark background. The prompt is [mohammadtofik@MOHAMMADTOFIK]~/. The command \$python3 index.py has been executed. The output shows "Array Creation :" followed by two blank lines, then "int64" and "float64" on separate lines.

```
Terminal
[mohammadtofik@MOHAMMADTOFIK]~/.
$python3 index.py
Array Creation :

int64
float64
```

```
print("\n\n")
print("Creating sequences of sequence will create 2 dimensional array : \n")
X = np.array([(3, 4, 5), (12, 6, 1)])
print(X)
Y = np.zeros((2, 4))
print(Y)
Z = np.ones((3, 3))
print(Z)
```

A terminal window showing the output of the previous code. It displays the message "Creating sequences of sequence will create 2 dimensional array :" followed by the output of X, Y, and Z. X is a 2x3 array with values [[3, 4, 5], [12, 6, 1]]. Y is a 2x4 array of zeros. Z is a 3x3 array of ones.

```
Creating sequences of sequence will create 2 dimensional array :

[[ 3  4  5]
 [12  6  1]]
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]]
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
```

```
print("To create a sequence of data : \n\n")
S = np.arange(10, 30, 5)
```

```
print(S)
P = np.arange(0, 2, 0.3)
print(P)
```

To create a sequence of data :

```
[10 15 20 25]
[0.  0.3 0.6 0.9 1.2 1.5 1.8]
```

```
print("\n\n")
print("Uses of Random Number function : \n")
print(random.choice([1, 2, 3, 4, 5]))
print(random.choice('python'))
print(random.randrange(25, 50))
print(random.randrange(25, 50, 2))
print(random.random())
print(random.uniform(5, 10))
print(random.shuffle([1, 2, 3, 4, 5]))
print(random.seed(10))
```

Uses of Random Number function :

```
4
p
35
25
0.8536134650561928
8.497284624448156
None
None
```

```
print("\n\n")
print("2-dimensional array (Matrix)")
a = np.arange(15).reshape(3, 5)
print(a)
print(a.shape)
print(a.size)
print(a.T)
```

```

2-dimensional array (Matrix)
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]]
(3, 5)
15
[[ 0  5 10]
 [ 1  6 11]
 [ 2  7 12]
 [ 3  8 13]
 [ 4  9 14]]

```

```

print("\n\n")
print("3-dimensional array : \n")
c = np.arange(24).reshape(2, 3, 4)
print(c)
print(c.shape)
print(c[1, :, :])
print(c[:, :, 2])

```

```

3-dimensional array :

[[[ 0  1  2  3]
   [ 4  5  6  7]
   [ 8  9 10 11]]

 [[12 13 14 15]
   [16 17 18 19]
   [20 21 22 23]]]
(2, 3, 4)
[[12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]]
[[ 2  6 10]
 [14 18 22]]

```

```

print("\n\n")
print("Array operation : \n")
a = np.array([20, 30, 40, 50])
b = np.arange(4)
print(b)
c = a-b
print(c)
print(b**2)
print(10*np.sin(a))
print(a < 35)

```

```

Array operation :

[0 1 2 3]
[20 29 38 47]
[0 1 4 9]
[ 9.12945251 -9.88031624  7.4511316  -2.62374854]
[ True  True False False]

```

```

print("\n\n")
print("Matrix operation : \n")
A = np.array([[1, 1], [0, 1]])
B = np.array([[2, 0], [3, 4]])
print(A*B)
print(A.dot(B))
print(np.dot(A, B))
C = np.arange(12).reshape(3, 4)
print(C)
print(C.sum(axis=0))
print(C.sum(axis=1))

```

```

Matrix operation :

[[2 0]
 [0 4]]
[[5 4]
 [3 4]]
[[5 4]
 [3 4]]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
[12 15 18 21]
[ 6 22 38]

```

```

print("\n\n")
print("Indexing, Slicing Litrating array : \n")
x = np.arange(10)**3
print(x)
print(x[2:5])
print(x[0:6:2])
y = np.array([[0, 1, 2, 3], [10, 11, 12, 13], [20, 21, 22, 23],
[30, 31, 32, 33], [40, 41, 42, 43]])
print(y[2, 3])

```

```

print(y[:5, 1])
print(y[-1, :])
print(y[:, -1])
for row in y:
    print(row)
for element in y.flat:
    print(element)

```

Indexing, Slicing Litrating array :

```

[ 0  1  8 27 64 125 216 343 512 729]
[ 8 27 64]
[ 0  8 64]
23
[ 1 11 21 31 41]
[40 41 42 43]
[ 3 13 23 33 43]
[0 1 2 3]
[10 11 12 13]
[20 21 22 23]
[30 31 32 33]
[40 41 42 43]
0
1
2
3
10
11
12
13
20
21

```

```

0
1
2
3
10
11
12
13
20
21
22
23
30
31
32
33
40
41
42
43

```

```

print("\n\n")
print("Creating the shape of the matrix : \n")
x = np.arange(10)**3
print(x)
print(x[2:5])
print(x[0:6:2])
y = np.array([[0, 1, 2, 3], [10, 11, 12, 13], [20, 21, 22, 23],
[30, 31, 32, 33], [40, 41, 42, 43]])
print(y[2, 3])
print(y[:5, 1])
print(y[-1, :])
print(y[:, -1])
for row in y:
print(row)
for element in y.flat:
print(element)
print(y.ravel())
B1 = y.reshape(4, 5)
print(B1)

```

```

Creating the shape of the matrix :

[  0   1   8  27  64 125 216 343 512 729]
[ 8 27 64]
[ 0  8 64]
23
[ 1 11 21 31 41]
[40 41 42 43]
[ 3 13 23 33 43]
[0 1 2 3]
[10 11 12 13]
[20 21 22 23]
[30 31 32 33]
[40 41 42 43]
0
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
1
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
2
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
3
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
10
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
11
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]

```

```

[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
22
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
23
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
30
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
31
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
32
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
33
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
40
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
41
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
42
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
43
[ 0  1  2  3 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43]
[[ 0  1  2  3 10]
 [11 12 13 20 21]
 [22 23 30 31 32]
 [33 40 41 42 43]]

```

```

print("\n\n")
print("Stacking together different array : \n")
A1 = np.array([(3, 4, 5), (12, 6, 1)])
A2 = np.array([(1, 2, 6), (-4, 3, 8)])
D1 = np.vstack((A1, A2))
D2 = np.hstack((A1, A2))
print(A1)
print(A2)
print(D1)
print(D2)

```

Stacking together different array :

```
[[ 3  4  5]
 [12  6  1]]
[[ 1  2  6]
 [-4  3  8]]
[[ 3  4  5]
 [12  6  1]
 [ 1  2  6]
 [-4  3  8]]
[[ 3  4  5  1  2  6]
 [12  6  1 -4  3  8]]
```

```
print("\n\n")
print("Stacking one D array into 2-D array : \n")
a = np.array([4., 2.])
b = np.array([3., 8.])
print(np.column_stack((a, b)))
print(np.hstack((a, b)))
```

Stacking one D array into 2-D array :

```
[[4. 3.]
 [2. 8.]]
[4. 2. 3. 8.]
```

```
print("\n\n")
print("Indexing with array of the indices : \n")
k = np.arange(12)**2
i = np.array([1, 1, 3, 8, 5])
j = np.array([[3, 4], [9, 7]])
print(k[i])
print(k[j])
```

Indexing with array of the indices :

```
[ 1  1  9 64 25]
[[ 9 16]
 [81 49]]
```



```

print("\n\n")
print("Use of for loop (Mapping by value : \n")
x = np.array([(3, 2, 9), (1, 6, 7)])
s1 = 0
for row in x:
    for col in row:
        s1 += col
print(s1)

print("\n\n")
print("Use of for loop (Mapping by Index : \n")
y = np.array([(3, 2, 9), (1, 6, 7)])
s = 0
for p in range(y.shape[0]):
    for q in range(y.shape[1]):
        s += y[p, q]
print(s)

```

```

Use of for loop (Mapping by value :

```

```

28

```

```

Use of for loop (Mapping by Index :

```

```

28

```

```

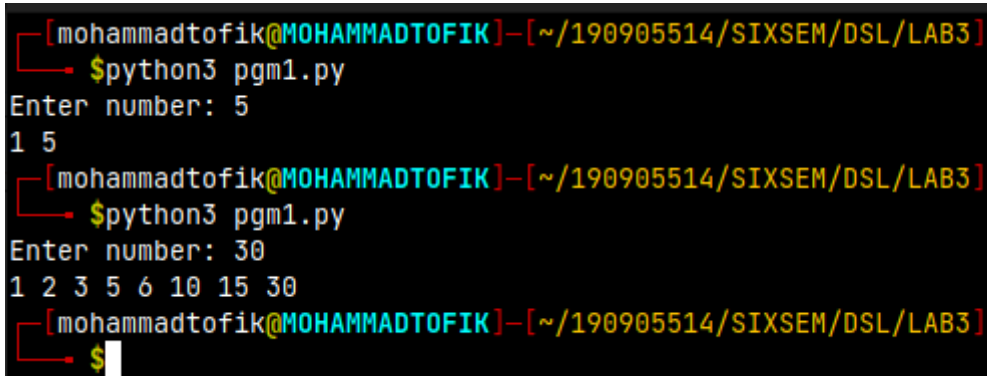
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$

```

## EXERCISE QUESTION :

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

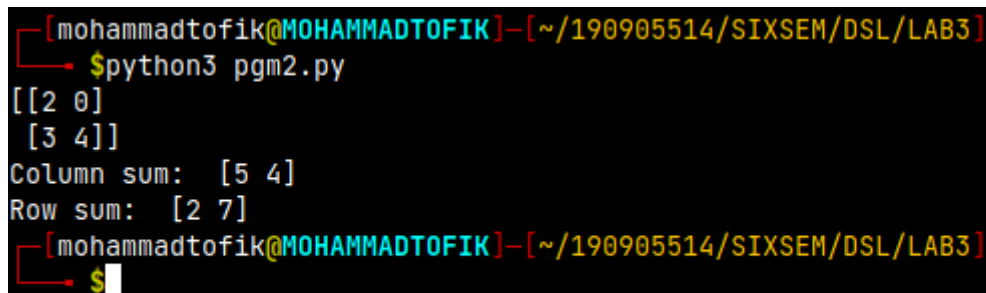
```
n = int(input("Enter number: "))
print('1', end=' ')
for i in range(2,n):
    if n%i == 0:
        print(i, end=' ')
print(n)
```



```
[mohammadtofik@MOHAMMADTOFIK]~/.ssh$ python3 pgm1.py
Enter number: 5
1 5
[mohammadtofik@MOHAMMADTOFIK]~/.ssh$ python3 pgm1.py
Enter number: 30
1 2 3 5 6 10 15 30
[mohammadtofik@MOHAMMADTOFIK]~/.ssh$
```

2. Find the sum of column and row using axis

```
import numpy as np
b = np.array([[2,0],[3,4]])
print(b)
print('Column sum: ', b.sum(axis=0))
print('Row sum: ', b.sum(axis=1))
```



```
[mohammadtofik@MOHAMMADTOFIK]~/.ssh$ python3 pgm2.py
[[2 0]
 [3 4]]
Column sum:  [5 4]
Row sum:  [2 7]
[mohammadtofik@MOHAMMADTOFIK]~/.ssh$
```

### 3. Operations on an array (use numpy wherever required):

(a) Create array from list with type float

(b) Create array from tuple

(c) Creating 3 X 4 array with all zeros;

(d) Creating a sequence of integers from 0 to 20 with step of 5

(e) Reshape 3 X 4 array to 2x2x3 array

(f) Find Maximum and minimum elements of an array, Row wise max and min, column wise max, and min and sum of elements. (Use function max(), min(), sum())

```
import numpy as np
# a)
a = np.array([1, 2, 3, 4, 5, 6])
print(a)
# b)
a = np.array((1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
print(a)
# c)
c = np.zeros((3, 4))
print(c)
# d)
d = np.arange(0, 20, 5)
print(d)
# e)
e = c.reshape(2, 2, 3)
print(e)
# f)
c = a.reshape(5, 2)
print(c)
print('Col min:', end=' ')
for i in range(c.shape[0]):
    print(min(c[i, :]), end=' ')
print('\nRow min:', end=' ')
for i in range(c.shape[1]):
    print(min(c[:, i]), end=' ')
print('Col max:', end=' ')
for i in range(c.shape[0]):
    print(max(c[i, :]), end=' ')
print('\nRow max:', end=' ')
for i in range(c.shape[1]):
    print(max(c[:, i]), end=' ')
print('\nColumn sum: ', c.sum(axis=0))
print('Row sum: ', c.sum(axis=1))
```

```

[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$python3 pgm3.py
[1 2 3 4 5 6]
[ 1 2 3 4 5 6 7 8 9 10]
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
[ 0 5 10 15]
[[[0. 0. 0.]
  [0. 0. 0.]]
 [[ 1 2]
 [ 3 4]
 [ 5 6]
 [ 7 8]
 [ 9 10]]]
Col min: 1 3 5 7 9
Row min: 1 2 Col max: 2 4 6 8 10
Row max: 9 10
Column sum: [25 30]
Row sum: [ 3 7 11 15 19]
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$

```

#### 4. Write a program to transpose of the matrix

```

import numpy as np
a = np.arange(0,10).reshape(2,5)
print(a)
a = a.T
print(a)

```

```

[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$python3 pgm4.py
[[0 1 2 3 4]
 [5 6 7 8 9]]
[[0 5]
 [1 6]
 [2 7]
 [3 8]
 [4 9]]
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$

```

### 5. Write a program to add two matrix

```
import numpy as np
a = np.arange(0,10).reshape(2,5)
b = np.arange(10,20).reshape(2,5)
print(a)
print(b)
print(a+b)
```

```
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$python3 pgm5.py
[[0 1 2 3 4]
 [5 6 7 8 9]]
[[10 11 12 13 14]
 [15 16 17 18 19]]
[[10 12 14 16 18]
 [20 22 24 26 28]]
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$
```

### 6. Write a program to find elements wise product between two matrices.

```
import numpy as np
print("Array Creation : ")
a = np.arange(0,10).reshape(2,5)
b = np.arange(10,20).reshape(2,5)
print(a)
print(b)
c = np.zeros((2,5))
for i in range(a.shape[0]):
    for j in range(a.shape[1]):c[i,j]=a[i,j]*b[i,j];
print(c)
```

```
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$python3 pgm6.py
Array Creation :
[[0 1 2 3 4]
 [5 6 7 8 9]]
[[10 11 12 13 14]
 [15 16 17 18 19]]
[[ 0.  11.  24.  39.  56.]
 [ 75.  96. 119. 144. 171.]]
[mohammadtofik@MOHAMMADTOFIK]--[~/190905514/SIXSEM/DSL/LAB3]
$
```