**Aarya Arban**

**S11-07**

## Assignment No. 15 – Linear Algebra using Scipy

## Code:

```
from scipy import constants
print(constants.yotta)
print(constants.zetta)
print(constants.exa)
print(constants.peta)
print(constants.tera)
print(constants.giga)
print(constants.mega)
print(constants.kilo)
print(constants.hecto)
print(constants.deka)
print(constants.deci)
print(constants.centi)
print(constants.milli)
print(constants.micro)
print(constants.nano)
print(constants.pico)
print(constants.femto)
print(constants.atto)
print(constants.zepto)
```

**Output:**

```
1e+24
```

1e+21

1e+18

1000000000000000.0

1000000000000.0

1000000000.0

1000000.0

1000.0

100.0

10.0

0.1

0.01

0.001

1e-06

1e-09

1e-12

1e-15

1e-18

1e-21

```
from scipy import constants
print(constants.kibi)
print(constants.mebi)
print(constants.gibi)
print(constants.tebi)
print(constants.pebi)
print(constants.exbi)
```

```
print(constants.zebi)
print(constants.yobi)
```

**Output:**

```
1024
1048576
1073741824
1099511627776
1125899906842624
1152921504606846976
1180591620717411303424
1208925819614629174706176
```

```
from scipy import constants
print(constants.degree)
print(constants.arcmin)
print(constants.arcminute)
print(constants.arcsec)
print(constants.arcsecond)
print(constants.minute)
print(constants.hour)
print(constants.day)
print(constants.week)
print(constants.year)
print(constants.Julian_year)
print(constants.inch)
print(constants.foot)
```

```
print(constants.yard)
print(constants.mile)
print(constants.mil)
print(constants.pt)
print(constants.point)
print(constants.survey_foot)
print(constants.survey_mile)
print(constants.nautical_mile)
print(constants.fermi)
print(constants.angstrom)
print(constants.micron)
print(constants.au)
print(constants.astronomical_unit)
print(constants.light_year)
print(constants.parsec)
```

**Output:**

```
0.017453292519943295
0.0002908882086657216
0.0002908882086657216
4.84813681109536e-06
4.84813681109536e-06
60.0
3600.0
86400.0
604800.0
31536000.0
```

31557600.0

0.0254

0.30479999999999996

0.9143999999999999

1609.3439999999998

2.5399999999999997e-05

0.00035277777777777776

0.00035277777777777776

0.3048006096012192

1609.3472186944373

1852.0

1e-15

1e-10

1e-06

149597870700.0

149597870700.0

9460730472580800.0

3.085677581491367e+16

```
from scipy.optimize import root
from math import cos
def eqn(x):
return x + cos(x)
myroot = root(eqn, 0)
print(myroot.x)
```

**Output:**

[-0.73908513]

```
from scipy import linalg
import numpy as np
a = np.array([[3, 2, 0], [1, -1, 0], [0, 5, 1]])
b = np.array([2, 4, -1])
x = linalg.solve(a, b)
print (x)
```

**Output:**

```
[ 2. -2. 9.]
```

```
from scipy import linalg
import numpy as np
A = np.array([[1,2],[3,4]])
x = linalg.det(A)
print (x)
```

**Output:**

```
-2.0
```

```
#importing the scipy and numpy packages
from scipy import linalg
import numpy as np
A = np.array([[1,2],[3,4]])
l, v = linalg.eig(A)
print (l)
print (v)
```

**Output:**

```
[-0.37228132+0.j 5.37228132+0.j]
[[-0.82456484 -0.41597356]
 [ 0.56576746 -0.90937671]]
```

```
array1 = np.array([1, 3, 5])
array2 = np.array([2, 4, 6])
result = np.dot(array1, array2)
print(result)
```

**Output:**

```
44
```

```
import numpy as np
array1 = np.array([[1, 3],[5, 7]])
array2 = np.array([[2, 4],[6, 8]])
result = np.inner(array1, array2)
print(result)
```

**Output:**

```
[[14 30]
 [38 86]]
```

```
import numpy as np
array1 = np.array([1, 3, 5])
array2 = np.array([2, 4, 6])
# outer() to perform outer multiplication
result = np.outer(array1, array2)
```

```
print(result)
```

**Output:**

```
[[ 2 4 6]
 [ 6 12 18]
 [10 20 30]]
```

```
import numpy as np
A = np.array([[2, 4],[6, 8]])
b = np.array([5, 6])
x = np.linalg.solve(A, b)
print(x)
```

**Output:**

```
[-2. 2.25]
[-2. 2.25]
```

```
import numpy as np
# Define a 3x3 matrix
array1 = np.array([[6, 3, 5],
                   [9, 2, 1],
                   [7, 8, 4]])
# Compute the trace of the matrix
result = np.trace(array1)
print(result)
```

**Output:**

```
12
```