AIM

1. Matrix operations (using vectorization) and transformation using python and SVD using Python.

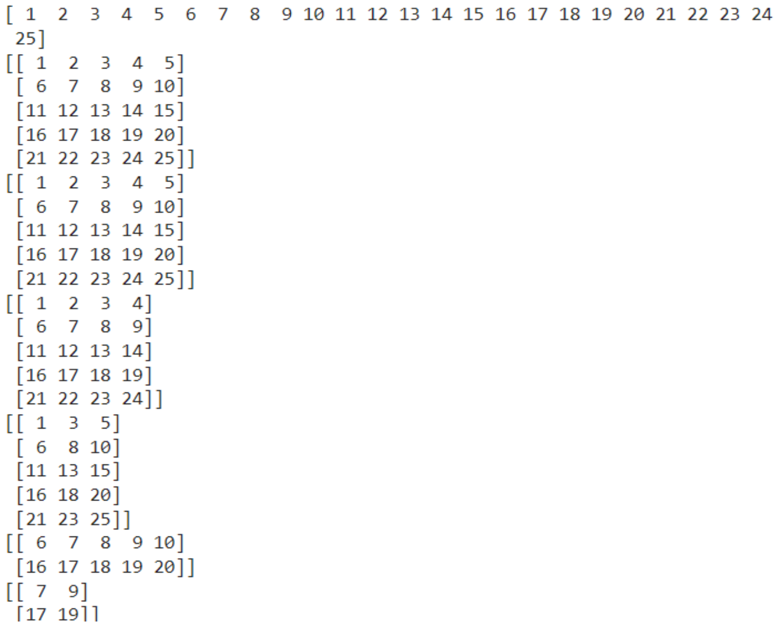
Programming Code:

```
 import numpy as np
y=np.arange(1,26)
print(y)

y=y.reshape(5,5)
print(y)

print(y[:5,:5])
print(y[:,:-1])
print(y[::,::2])
print(y[1::2,::])
print(y[1::2,1::2])
```

OUTPUT:

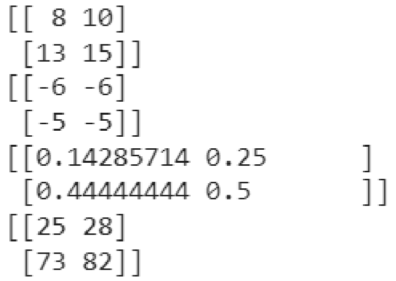


```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25]
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]
 [21 22 23 24 25]]
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]
 [21 22 23 24 25]]
[[ 1  2  3  4]
 [ 6  7  8  9]
 [11 12 13 14]
 [16 17 18 19]
 [21 22 23 24]]
[[ 1  3  5]
 [ 6  8 10]
 [11 13 15]
 [16 18 20]
 [21 23 25]]
[[ 6  7  8  9 10]
 [16 17 18 19 20]]
[[ 7  9]
 [17 19]]
```

Programming Code:

```
import numpy
x=numpy.array([[1,2],[4,5]])
y=numpy.array([[7,8],[9,10]])
print(np.add(x,y))
print(np.subtract(x,y))
print(np.divide(x,y))
print(np.dot(x,y))
```

OUTPUT:

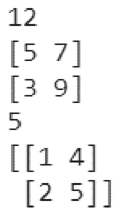


```
[[ 8 10]
 [13 15]]
[[-6 -6]
 [-5 -5]]
[[0.14285714 0.25      ]
 [0.44444444 0.5       ]]
[[25 28]
 [73 82]]
```

Programming Code:

```
print(x.sum())
print(x.sum(axis=0))
print(x.sum(axis=1))
print(x.max())
print(x.transpose())
```

OUTPUT:

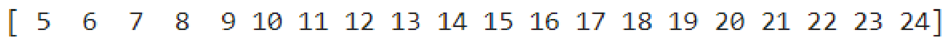


```
12
[5 7]
[3 9]
5
[[1 4]
 [2 5]]
```

Programming Code:

```
print(y[4:-1])
```

OUTPUT:



```
[ 5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]
```