```
import numpy as np
```

1) Define a NumPy array to represent the vector $v = [1, 5, 2]$.

```
v = np.array([1, 5, 2])
v
```

```
array([1, 5, 2])
```

2) Define a NumPy array to represent the vector $w = [0, ..., 8]$.

```
w = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8])
w
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
```

3) Define a NumPy array to represent the matrix

$$A = \begin{bmatrix} 2 & 5 & 6 \\ 3 & 2 & 1 \\ 4 & 9 & 3 \end{bmatrix}$$

```
A = np.array([[2, 5, 6],
              [3, 2, 1],
              [4, 9, 3]])
A
```

```
array([[2, 5, 6],
       [3, 2, 1],
       [4, 9, 3]])
```

4) Define a NumPy array to represent the matrix

$$B = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

```
B = np.array([[0, 1, 2],
              [3, 4, 5],
              [6, 7, 8]])
B
```

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

5) Extract the third element of v.

```
v[2]
```

```
2
```

6) Extract the second, third, and fourth elements of w.

```
w[1:4]
```

```
array([1, 2, 3])
```

7) Extract the element in the first row and the second column of A.

```
A[0, 1]
```

```
5
```

8) Extract the second column of B.

```
B[:, 1]
```

```
array([1, 4, 7])
```

9) Calculate the transpose of A.

```
A.T
```

```
array([[2, 3, 4],
       [5, 2, 9],
       [6, 1, 3]])
```

10) Calculate determinant of A (using the function `linalg.det`)

```
np.linalg.det(A)
```

```
82.99999999999996
```

11) Calculate the inverse of A.

```
np.linalg.inv(A)
```

```
array([[-0.03614458,  0.46987952, -0.08433735],
       [-0.06024096, -0.21686747,  0.19277108],
       [ 0.22891566,  0.02409639, -0.13253012]])
```

12) Calculate Av.

```
np.dot(A, v)
```

```
array([39, 15, 55])
```

13) Calculate the matrix product AB.

```
np.dot(A, B)
```

```
array([[51, 64, 77],
       [12, 18, 24],
       [45, 61, 77]])
```

14) Caclulate $3A^2 + 2A$.

```
A_squared = np.dot(A, A)
3 * A_squared + 2 * A
```

```
array([[133, 232, 117],
       [ 54,  88,  71],
       [149, 213, 132]])
```

15) Solve the system of linear equations $Ax = v$ for x.

```
x = np.linalg.solve(A, v)
x
```

```
array([ 2.14457831, -0.75903614,  0.08433735])
```