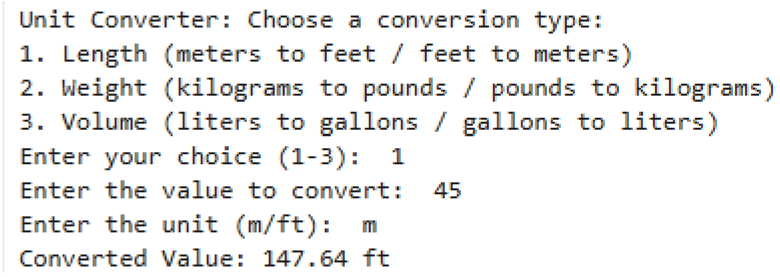
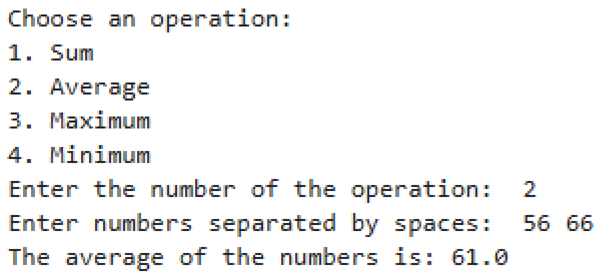
Output of Task1



```
Unit Converter: Choose a conversion type:
1. Length (meters to feet / feet to meters)
2. Weight (kilograms to pounds / pounds to kilograms)
3. Volume (liters to gallons / gallons to liters)
Enter your choice (1-3):  1
Enter the value to convert:  45
Enter the unit (m/ft):  m
Converted Value: 147.64 ft
```

Output of Task2



```
Choose an operation:
1. Sum
2. Average
3. Maximum
4. Minimum
Enter the number of the operation:  2
Enter numbers separated by spaces:  56 66
The average of the numbers is: 61.0
```

Output of Task3

```
[1, 3, 5]
```

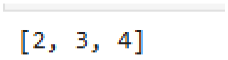
Output of Task4

```
[3, 4, 5]
```

Output of Task5

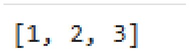
```
[5, 4, 3, 2, 1]
```

Output of Task6



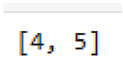
```
[2, 3, 4]
```

Task 7



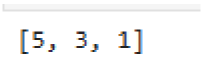
```
[1, 2, 3]
```

Task8



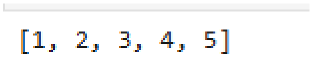
```
[4, 5]
```

Task 9



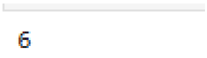
```
[5, 3, 1]
```

Task10



```
[1, 2, 3, 4, 5]
```

Task11

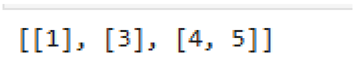


```
6
```

Task12

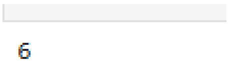
```
21
```

Task13



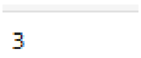
```
[[1], [3], [4, 5]]
```

Task14



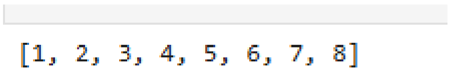
```
6
```

Task15



```
3
```

Task 16

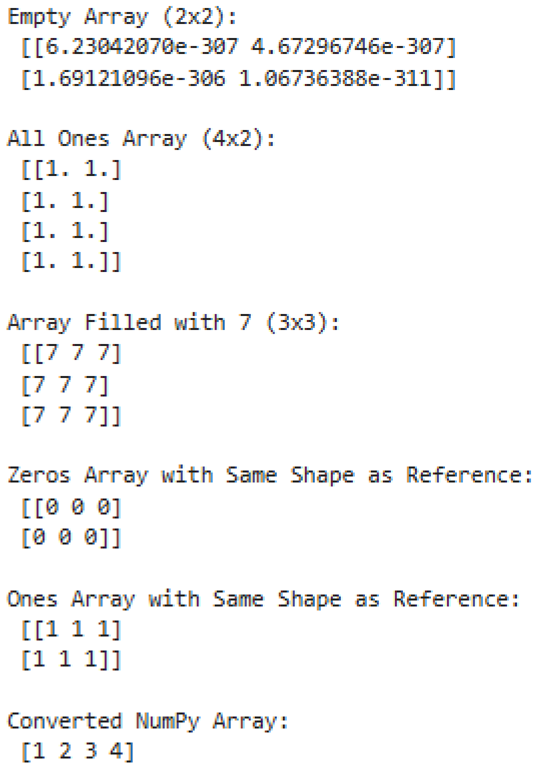


```
[1, 2, 3, 4, 5, 6, 7, 8]
```

Task 17

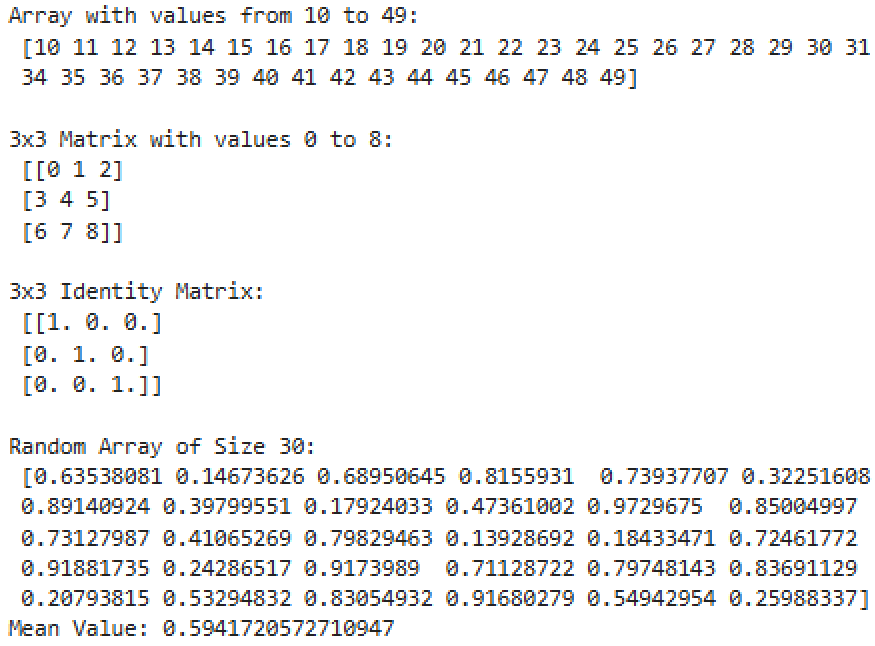
```
3.5
```

Task18

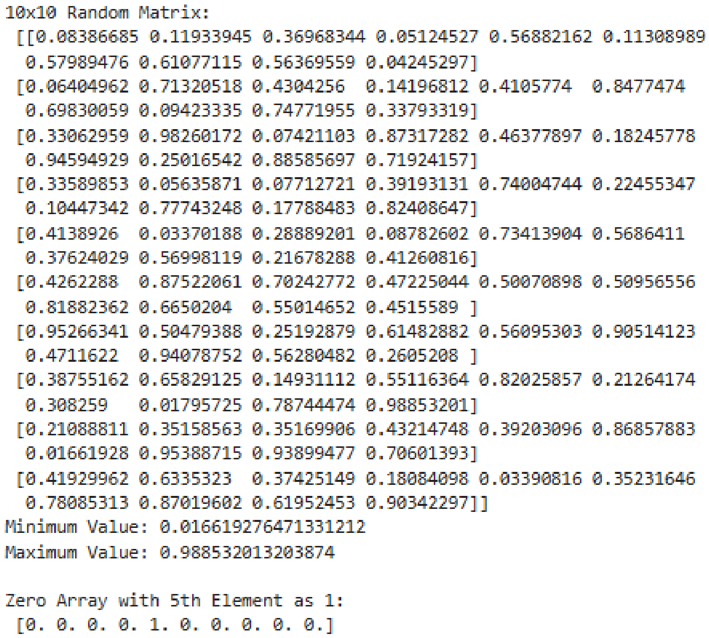


```
Empty Array (2x2):
 [[6.23042070e-307 4.67296746e-307]
 [1.69121096e-306 1.06736388e-311]]

All Ones Array (4x2):
 [[1. 1.]
 [1. 1.]
 [1. 1.]
 [1. 1.]]

Array Filled with 7 (3x3):
 [[7 7 7]
 [7 7 7]
 [7 7 7]]

Zeros Array with Same Shape as Reference:
 [[0 0 0]
 [0 0 0]]

Ones Array with Same Shape as Reference:
 [[1 1 1]
 [1 1 1]]

Converted NumPy Array:
 [1 2 3 4]
```

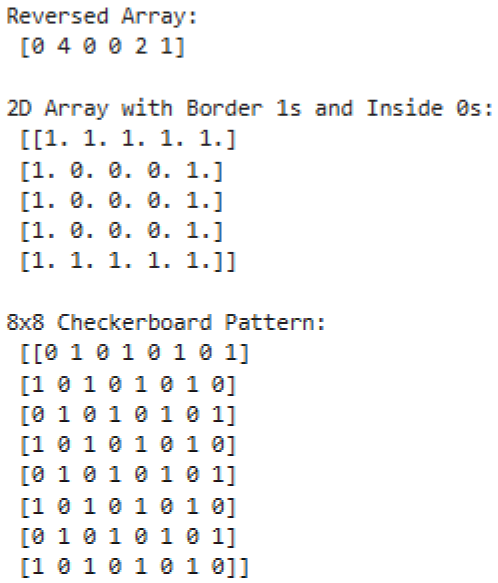
Task19



```
Array with values from 10 to 49:
 [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49]

3x3 Matrix with values 0 to 8:
 [[0 1 2]
 [3 4 5]
 [6 7 8]]

3x3 Identity Matrix:
 [[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

Random Array of Size 30:
 [0.63538081 0.14673626 0.68950645 0.8155931  0.73937707 0.32251608
 0.89140924 0.39799551 0.17924033 0.47361002 0.9729675  0.85004997
 0.73127987 0.41065269 0.79829463 0.13928692 0.18433471 0.72461772
 0.91881735 0.24286517 0.9173989  0.71128722 0.79748143 0.83691129
 0.20793815 0.53294832 0.83054932 0.91680279 0.54942954 0.25988337]
Mean Value: 0.5941720572710947
```

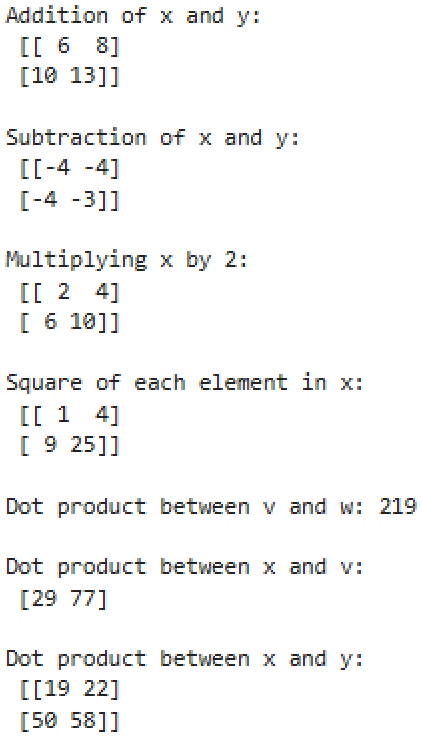


```
10x10 Random Matrix:
 [[0.08386685 0.11933945 0.36968344 0.05124527 0.56882162 0.11308989
  0.57989476 0.61077115 0.56369559 0.04245297]
 [0.06404962 0.71320518 0.4304256  0.14196812 0.4105774  0.8477474
  0.69830059 0.09423335 0.74771955 0.33793319]
 [0.33062959 0.98260172 0.07421103 0.87317282 0.46377897 0.18245778
  0.94594929 0.25016542 0.88585697 0.71924157]
 [0.33589853 0.05635871 0.07712721 0.39193131 0.74004744 0.22455347
  0.10447342 0.77743248 0.17788483 0.82408647]
 [0.4138926  0.03370188 0.28889201 0.08782602 0.73413904 0.5686411
  0.37624029 0.56998119 0.21678288 0.41260816]
 [0.4262288  0.87522061 0.70242772 0.47225044 0.50070898 0.50956556
  0.81882362 0.6650204  0.55014652 0.4515589 ]
 [0.95266341 0.50479388 0.25192879 0.61482882 0.56095303 0.90514123
  0.4711622  0.94078752 0.56280482 0.2605208 ]
 [0.38755162 0.65829125 0.14931112 0.55116364 0.82025857 0.21264174
  0.308259   0.01795725 0.78744474 0.98853201]
 [0.21088811 0.35158563 0.35169906 0.43214748 0.39203096 0.86857883
  0.01661928 0.95388715 0.93899477 0.70601393]
 [0.41929962 0.6335323  0.37425149 0.18084098 0.03390816 0.35231646
  0.78085313 0.87019602 0.61952453 0.90342297]]
Minimum Value: 0.016619276471331212
Maximum Value: 0.988532013203874

Zero Array with 5th Element as 1:
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
```

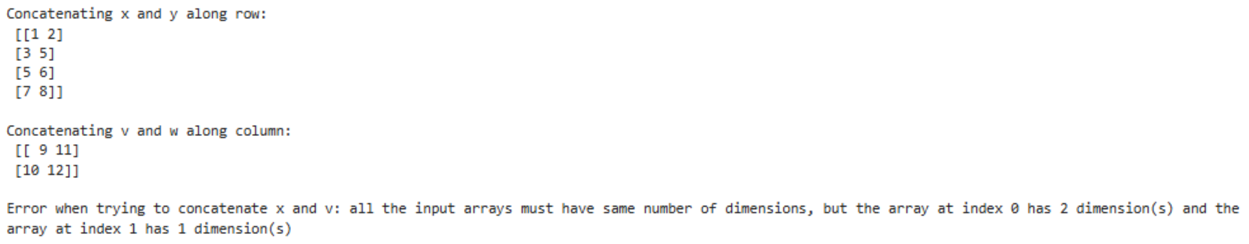


```
Reversed Array:
 [0 4 0 0 2 1]

2D Array with Border 1s and Inside 0s:
 [[1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1.]]

8x8 Checkerboard Pattern:
 [[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]]
```

Task20

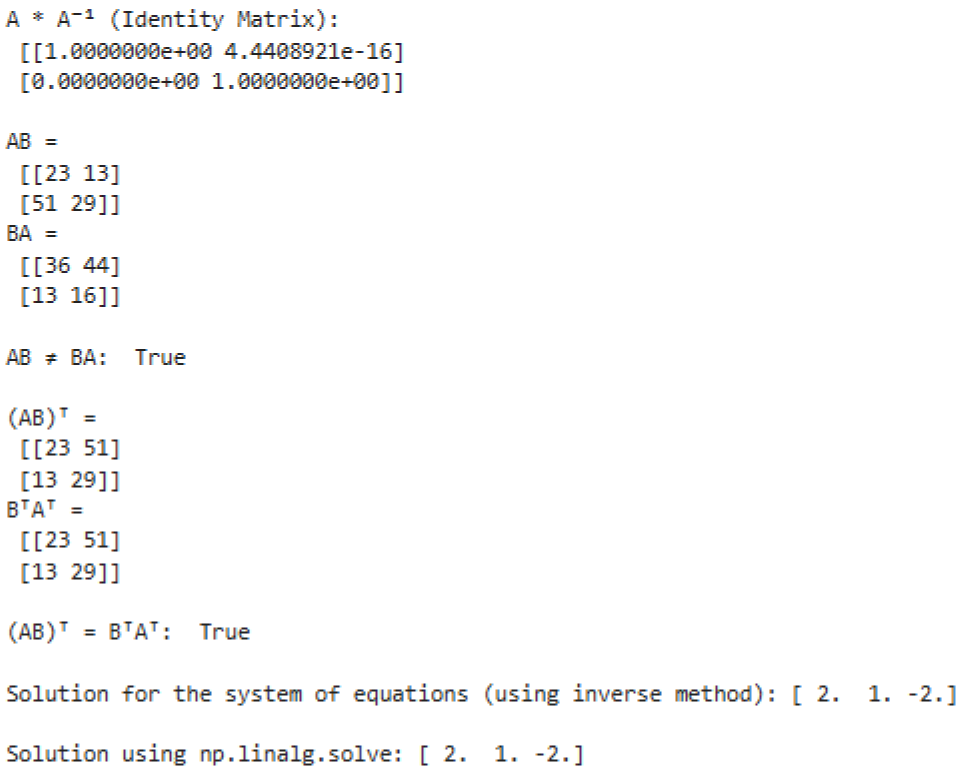


```
Addition of x and y:
 [[ 6  8]
 [10 13]]

Subtraction of x and y:
 [[-4 -4]
 [-4 -3]]

Multiplying x by 2:
 [[ 2  4]
 [ 6 10]]

Square of each element in x:
 [[ 1  4]
 [ 9 25]]

Dot product between v and w: 219

Dot product between x and v:
 [29 77]

Dot product between x and y:
 [[19 22]
 [50 58]]
```



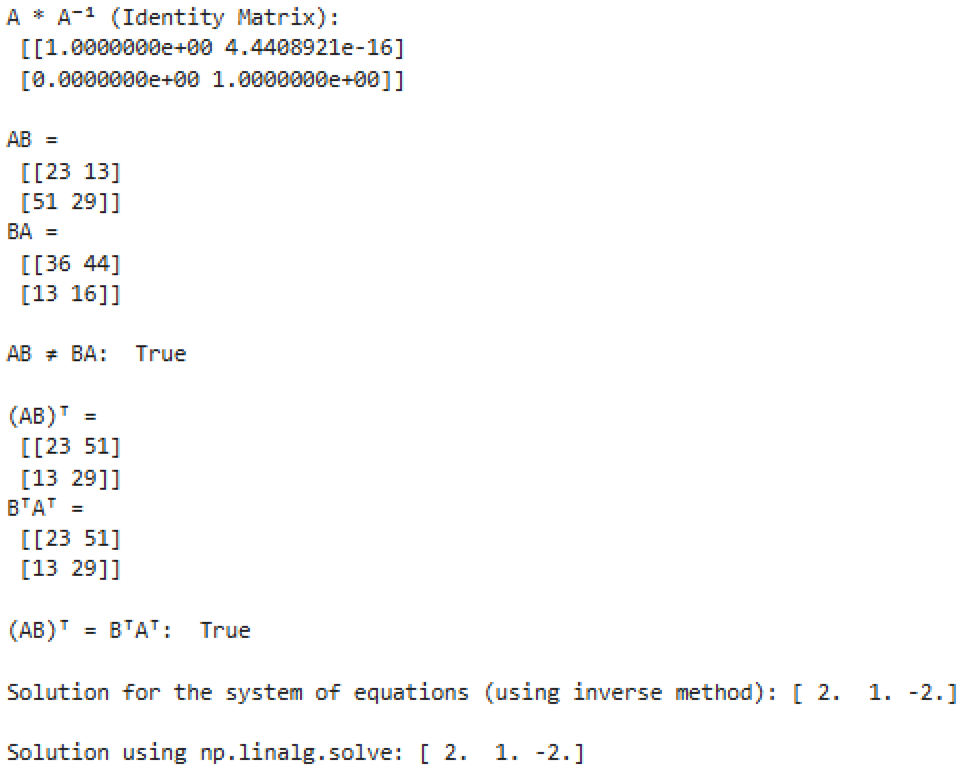
```
Concatenating x and y along row:
 [[1 2]
 [3 5]
 [5 6]
 [7 8]]

Concatenating v and w along column:
 [[ 9 11]
 [10 12]]

Error when trying to concatenate x and v: all the input arrays must have same number of dimensions, but the array at index 0 has 2 dimension(s) and the
array at index 1 has 1 dimension(s)
```

Task 21



```
A * A⁻¹ (Identity Matrix):
 [[1.0000000e+00 4.4408921e-16]
 [0.0000000e+00 1.0000000e+00]]

AB =
 [[23 13]
 [51 29]]
BA =
 [[36 44]
 [13 16]]

AB ≠ BA:  True

(AB)ᵀ =
 [[23 51]
 [13 29]]
BᵀAᵀ =
 [[23 51]
 [13 29]]

(AB)ᵀ = BᵀAᵀ:  True

Solution for the system of equations (using inverse method): [ 2.  1. -2.]

Solution using np.linalg.solve: [ 2.  1. -2.]
```

Task 22



```
A * A⁻¹ (Identity Matrix):
 [[1.0000000e+00 4.4408921e-16]
 [0.0000000e+00 1.0000000e+00]]

AB =
 [[23 13]
 [51 29]]
BA =
 [[36 44]
 [13 16]]

AB ≠ BA:  True

(AB)ᵀ =
 [[23 51]
 [13 29]]
BᵀAᵀ =
 [[23 51]
 [13 29]]

(AB)ᵀ = BᵀAᵀ:  True

Solution for the system of equations (using inverse method): [ 2.  1. -2.]

Solution using np.linalg.solve: [ 2.  1. -2.]
```