

In [31]:

```
import numpy as np
```

In [ ]:

In [5]:

```
a = np.loadtxt("testmarks1.csv",skiprows = 1,delimiter=',')
b = np.loadtxt("testmarks2.csv",skiprows = 1,delimiter=',')
print(a)
print("\n")
print(b)
```

```
[[43.05 27.79 28.7 27.79]
 [43.47 28.52 28.98 27.89]
 [42.24 28.16 28.16 25.63]
 [39.24 26.16 26.16 26.16]
 [40.9 26.03 27.27 25.65]
 [39.47 26.31 26.31 25.21]
 [41.68 25.63 27.79 25.46]
 [42.19 27.61 28.13 26.21]
 [44.75 28.35 29.83 28.21]
 [46.95 28.88 31.3 28.53]]
```

```
[[28.48 34.18 30.56 22.23]
 [28.1 33.72 30.68 22.82]
 [26.16 31.39 28.2 22.53]
 [26.16 31.39 28.78 20.93]
 [26.1 31.32 28.22 20.82]
 [25.45 30.54 27.73 21.05]
 [26.16 31.39 28.01 20.51]
 [27.44 32.93 28.83 22.08]
 [28.63 34.35 31.03 22.68]
 [30.35 36.42 31.38 23.1 ]]
```

In [6]:

```
add = np.add(a,b)
print("Sum of the two marksheets is \n\n",add)
```

Sum of the two marksheets is

```
[[71.53 61.97 59.26 50.02]
 [71.57 62.24 59.66 50.71]
 [68.4  59.55 56.36 48.16]
 [65.4  57.55 54.94 47.09]
 [67.    57.35 55.49 46.47]
 [64.92 56.85 54.04 46.26]
 [67.84 57.02 55.8  45.97]
 [69.63 60.54 56.96 48.29]
 [73.38 62.7  60.86 50.89]
 [77.3  65.3  62.68 51.63]]
```

In [7]:

```
subs = np.subtract(a,b)
print("Substraction of the two marksheets is \n\n ",subs)
```

Substraction of the two marksheets is

```
[[14.57 -6.39 -1.86  5.56]
 [15.37 -5.2  -1.7   5.07]
 [16.08 -3.23 -0.04  3.1  ]
 [13.08 -5.23 -2.62  5.23]
 [14.8  -5.29 -0.95  4.83]
 [14.02 -4.23 -1.42  4.16]
 [15.52 -5.76 -0.22  4.95]
 [14.75 -5.32 -0.7   4.13]
 [16.12 -6.    -1.2   5.53]
 [16.6  -7.54 -0.08  5.43]]
```

In [8]:

```
multi = np.multiply(a,b)
print("Multiplication of the two marksheets is \n\n",multi)
```

Multiplication of the two marksheets is

```
[[1226.064  949.8622  877.072  617.7717]
 [1221.507  961.6944  889.1064  636.4498]
 [1104.9984  883.9424  794.112  577.4439]
 [1026.5184  821.1624  752.8848  547.5288]
 [1067.49    815.2596  769.5594  534.033 ]
 [1004.5115  803.5074  729.5763  530.6705]
 [1090.3488  804.5257  778.3979  522.1846]
 [1157.6936  909.1973  810.9879  578.7168]
 [1281.1925  973.8225  925.6249  639.8028]
 [1424.9325 1051.8096  982.194  659.043  ]]
```

In [9]:

```
div = np.divide(a,b)
print("Division of the two marksheets is\n\n",div)
```

Division of the two marksheets is

```
[[1.51158708 0.81304857 0.93913613 1.25011246]
 [1.54697509 0.84578885 0.94458931 1.22217353]
 [1.6146789  0.89710099 0.99858156 1.13759432]
 [1.5         0.83338643 0.90896456 1.24988055]
 [1.56704981 0.83109834 0.96633593 1.23198847]
 [1.55088409 0.86149312 0.94879192 1.1976247 ]
 [1.59327217 0.81650207 0.99214566 1.24134569]
 [1.53753644 0.83844519 0.97571974 1.1870471 ]
 [1.56304576 0.82532751 0.96132775 1.24382716]
 [1.54695222 0.7929709  0.99745061 1.23506494]]
```

In [10]:

```
mean = np.mean(a)
print("Mean of the testsheet1 is",mean,"\n\n")
mean2 = np.mean(b)
print("Mean of the testsheet2 is",mean2,"\n\n")
```

Mean of the testsheet1 is 31.16875

Mean of the testsheet2 is 27.820749999999997

In [11]:

```
power = a**2
print(power)
power2 = np.square(a)
print(power2)
```

```
[[1853.3025  772.2841  823.69    772.2841]
 [1889.6409  813.3904  839.8404  777.8521]
 [1784.2176  792.9856  792.9856  656.8969]
 [1539.7776  684.3456  684.3456  684.3456]
 [1672.81    677.5609  743.6529  657.9225]
 [1557.8809  692.2161  692.2161  635.5441]
 [1737.2224  656.8969  772.2841  648.2116]
 [1779.9961  762.3121  791.2969  686.9641]
 [2002.5625  803.7225  889.8289  795.8041]
 [2204.3025  834.0544  979.69    813.9609]]
[[1853.3025  772.2841  823.69    772.2841]
 [1889.6409  813.3904  839.8404  777.8521]
 [1784.2176  792.9856  792.9856  656.8969]
 [1539.7776  684.3456  684.3456  684.3456]
 [1672.81    677.5609  743.6529  657.9225]
 [1557.8809  692.2161  692.2161  635.5441]
 [1737.2224  656.8969  772.2841  648.2116]
 [1779.9961  762.3121  791.2969  686.9641]
 [2002.5625  803.7225  889.8289  795.8041]
 [2204.3025  834.0544  979.69    813.9609]]
```

In [12]:

```
transpose = a.T
print(transpose)
```

```
[[43.05 43.47 42.24 39.24 40.9  39.47 41.68 42.19 44.75 46.95]
 [27.79 28.52 28.16 26.16 26.03 26.31 25.63 27.61 28.35 28.88]
 [28.7  28.98 28.16 26.16 27.27 26.31 27.79 28.13 29.83 31.3 ]
 [27.79 27.89 25.63 26.16 25.65 25.21 25.46 26.21 28.21 28.53]]
```

In [13]:

```
horizontal_stack = np.hstack((a,b))
print(horizontal_stack,"Horizontal stack \n\n ")

vertical_stack = np.vstack((a,b))
print(vertical_stack,"vertical stack \n\n ")
```

```
[[43.05 27.79 28.7 27.79 28.48 34.18 30.56 22.23]
 [43.47 28.52 28.98 27.89 28.1 33.72 30.68 22.82]
 [42.24 28.16 28.16 25.63 26.16 31.39 28.2 22.53]
 [39.24 26.16 26.16 26.16 26.16 31.39 28.78 20.93]
 [40.9 26.03 27.27 25.65 26.1 31.32 28.22 20.82]
 [39.47 26.31 26.31 25.21 25.45 30.54 27.73 21.05]
 [41.68 25.63 27.79 25.46 26.16 31.39 28.01 20.51]
 [42.19 27.61 28.13 26.21 27.44 32.93 28.83 22.08]
 [44.75 28.35 29.83 28.21 28.63 34.35 31.03 22.68]
 [46.95 28.88 31.3 28.53 30.35 36.42 31.38 23.1 ]] Horizontal stack
```

```
[[43.05 27.79 28.7 27.79]
 [43.47 28.52 28.98 27.89]
 [42.24 28.16 28.16 25.63]
 [39.24 26.16 26.16 26.16]
 [40.9 26.03 27.27 25.65]
 [39.47 26.31 26.31 25.21]
 [41.68 25.63 27.79 25.46]
 [42.19 27.61 28.13 26.21]
 [44.75 28.35 29.83 28.21]
 [46.95 28.88 31.3 28.53]
 [28.48 34.18 30.56 22.23]
 [28.1 33.72 30.68 22.82]
 [26.16 31.39 28.2 22.53]
 [26.16 31.39 28.78 20.93]
 [26.1 31.32 28.22 20.82]
 [25.45 30.54 27.73 21.05]
 [26.16 31.39 28.01 20.51]
 [27.44 32.93 28.83 22.08]
 [28.63 34.35 31.03 22.68]
 [30.35 36.42 31.38 23.1 ]] vertical stack
```

In [14]:

```
custom_sequence = np.arange(69,420,96)
print(custom_sequence)
```

```
[ 69 165 261 357]
```

In [15]:

```
std_deviation = np.std(a)
print(std_deviation)
```

```
6.692269864365901
```

In [16]:

```
minimum = np.min(a)
maximum = np.max(a)
print(minimum, "\n\n", maximum)
```

25.21

46.95

In [17]:

```
copy_array = a.copy()
print(copy_array)
```

```
[[43.05 27.79 28.7 27.79]
 [43.47 28.52 28.98 27.89]
 [42.24 28.16 28.16 25.63]
 [39.24 26.16 26.16 26.16]
 [40.9 26.03 27.27 25.65]
 [39.47 26.31 26.31 25.21]
 [41.68 25.63 27.79 25.46]
 [42.19 27.61 28.13 26.21]
 [44.75 28.35 29.83 28.21]
 [46.95 28.88 31.3 28.53]]
```

In [18]:

```
#search
index = np.where(a == 42.19)
print(index)
print(type(index))
```

```
(array([7], dtype=int64), array([0], dtype=int64))
<class 'tuple'>
```

In [19]:

```
#sort
sorted_data = np.sort(a)
print(sorted_data)
```

```
[[27.79 27.79 28.7 43.05]
 [27.89 28.52 28.98 43.47]
 [25.63 28.16 28.16 42.24]
 [26.16 26.16 26.16 39.24]
 [25.65 26.03 27.27 40.9 ]
 [25.21 26.31 26.31 39.47]
 [25.46 25.63 27.79 41.68]
 [26.21 27.61 28.13 42.19]
 [28.21 28.35 29.83 44.75]
 [28.53 28.88 31.3 46.95]]
```

In [20]:

```
broadcasting, counts = np.unique(a[:,1], return_counts= True)
print(broadcasting)
```

```
[25.63 26.03 26.16 26.31 27.61 27.79 28.16 28.35 28.52 28.88]
```

In [21]:

```
modarray = np.mod(a,b)
print(modarray)
```

```
[[14.57 27.79 28.7  5.56]
 [15.37 28.52 28.98 5.07]
 [16.08 28.16 28.16 3.1 ]
 [13.08 26.16 26.16 5.23]
 [14.8  26.03 27.27 4.83]
 [14.02 26.31 26.31 4.16]
 [15.52 25.63 27.79 4.95]
 [14.75 27.61 28.13 4.13]
 [16.12 28.35 29.83 5.53]
 [16.6  28.88 31.3  5.43]]
```

In [22]:

```
shape = a.shape
print(shape)
```

```
(10, 4)
```

In [30]:

```
print("Original \n",a,"\n\n")
reshape = a.reshape(4,10)
print("Reshaped \n",reshape)
```

Original

```
[[43.05 27.79 28.7  27.79]
 [43.47 28.52 28.98 27.89]
 [42.24 28.16 28.16 25.63]
 [39.24 26.16 26.16 26.16]
 [40.9  26.03 27.27 25.65]
 [39.47 26.31 26.31 25.21]
 [41.68 25.63 27.79 25.46]
 [42.19 27.61 28.13 26.21]
 [44.75 28.35 29.83 28.21]
 [46.95 28.88 31.3  28.53]]
```

Reshaped

```
[[43.05 27.79 28.7  27.79 43.47 28.52 28.98 27.89 42.24 28.16]
 [28.16 25.63 39.24 26.16 26.16 26.16 40.9  26.03 27.27 25.65]
 [39.47 26.31 26.31 25.21 41.68 25.63 27.79 25.46 42.19 27.61]
 [28.13 26.21 44.75 28.35 29.83 28.21 46.95 28.88 31.3  28.53]]
```

In [ ]: