Results:

Matrix M of size 20 x 20.

Matrix M of size 20×20 with entries given as M(i,j) = (1/2)*(i - j), i, j = 0, ..., 19

0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000	-7.0000	-7.5000	-8.0000	-8.5000	-9.0000	-9.5000
0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000	-7.0000	-7.5000	-8.0000	-8.5000	-9.0000
1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000	-7.0000	-7.5000	-8.0000	-8.5000
1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000	-7.0000	-7.5000	-8.0000
2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000	-7.0000	-7.5000
2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000	-7.0000
3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000	-6.5000
3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000	-6.0000
4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000	-5.5000
4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000	-5.0000
5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000	-4.5000
5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000	-4.0000
6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000	-3.5000
6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000	-3.0000
7.0000	6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000	-2.5000
7.5000	7.0000	6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000	-2.0000
8.0000	7.5000	7.0000	6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000	-1.5000
8.5000	8.0000	7.5000	7.0000	6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000	-1.0000
9.0000	8.5000	8.0000	7.5000	7.0000	6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0	-0.5000
9.5000	9.0000	8.5000	8.0000	7.5000	7.0000	6.5000	6.0000	5.5000	5.0000	4.5000	4.0000	3.5000	3.0000	2.5000	2.0000	1.5000	1.0000	0.5000	0

1. Testing function Code 1:

```
1.1   Trace of the Matrix M    = 0.000000
1.2.1 Sum of the entries       = 0.000000
1.2.2 Mean of the entries      = 0.000000
1.2.3 Maximum of the entries   = 9.500000
1.3.1 Sum of the absolute values of entries      = 1330.000000
1.3.2 Mean of the absolute values of entries     = 3.325000
1.3.3 Maximum of the absolute values of entries  = 9.500000
```

1.4 A vector of size N × 1 = 20 x 1 containing the mean of each row.

```
-4.7500
-4.2500
-3.7500
-3.2500
-2.7500
-2.2500
-1.7500
-1.2500
-0.7500
-0.2500
 0.2500
 0.7500
 1.2500
 1.7500
 2.2500
 2.7500
 3.2500
 3.7500
 4.2500
 4.7500
```

1.5 A vector of size N × 1 = 20 x 1 containing the mean of each column

```
 4.7500
 4.2500
 3.7500
 3.2500
 2.7500
 2.2500
 1.7500
 1.2500
 0.7500
 0.2500
-0.2500
-0.7500
-1.2500
-1.7500
-2.2500
-2.7500
-3.2500
-3.7500
-4.2500
-4.7500
```

M * M transpose

1.6 A matrix of size N × N containing the result of multiplying M with its transpose.

```
  617.5000  570.0000  522.5000  475.0000  427.5000  380.0000  332.5000  285.0000  237.5000  190.0000  142.5000   95.0000   47.5000         0  -47.5000  -95.0000 -142.5000 -190.0000 -237.5000 -285.0000
  570.0000  527.5000  485.0000  442.5000  400.0000  357.5000  315.0000  272.5000  230.0000  187.5000  145.0000  102.5000   60.0000   17.5000  -25.0000  -67.5000 -110.0000 -152.5000 -195.0000 -237.5000
  522.5000  485.0000  447.5000  410.0000  372.5000  335.0000  297.5000  260.0000  222.5000  185.0000  147.5000  110.0000   72.5000   35.0000   -2.5000  -40.0000  -77.5000 -115.0000 -152.5000 -190.0000
  475.0000  442.5000  410.0000  377.5000  345.0000  312.5000  280.0000  247.5000  215.0000  182.5000  150.0000  117.5000   85.0000   52.5000   20.0000  -12.5000  -45.0000  -77.5000 -110.0000 -142.5000
  427.5000  400.0000  372.5000  345.0000  317.5000  290.0000  262.5000  235.0000  207.5000  180.0000  152.5000  125.0000   97.5000   70.0000   42.5000   15.0000  -12.5000  -40.0000  -67.5000  -95.0000
  380.0000  357.5000  335.0000  312.5000  290.0000  267.5000  245.0000  222.5000  200.0000  177.5000  155.0000  132.5000  110.0000   87.5000   65.0000   42.5000   20.0000   -2.5000  -25.0000  -47.5000
  332.5000  315.0000  297.5000  280.0000  262.5000  245.0000  227.5000  210.0000  192.5000  175.0000  157.5000  140.0000  122.5000  105.0000   87.5000   70.0000   52.5000   35.0000   17.5000         0
  285.0000  272.5000  260.0000  247.5000  235.0000  222.5000  210.0000  197.5000  185.0000  172.5000  160.0000  147.5000  135.0000  122.5000  110.0000   97.5000   85.0000   72.5000   60.0000   47.5000
  237.5000  230.0000  222.5000  215.0000  207.5000  200.0000  192.5000  185.0000  177.5000  170.0000  162.5000  155.0000  147.5000  140.0000  132.5000  125.0000  117.5000  110.0000  102.5000   95.0000
  190.0000  187.5000  185.0000  182.5000  180.0000  177.5000  175.0000  172.5000  170.0000  167.5000  165.0000  162.5000  160.0000  157.5000  155.0000  152.5000  150.0000  147.5000  145.0000  142.5000
  142.5000  145.0000  147.5000  150.0000  152.5000  155.0000  157.5000  160.0000  162.5000  165.0000  167.5000  170.0000  172.5000  175.0000  177.5000  180.0000  182.5000  185.0000  187.5000  190.0000
   95.0000  102.5000  110.0000  117.5000  125.0000  132.5000  140.0000  147.5000  155.0000  162.5000  170.0000  177.5000  185.0000  192.5000  200.0000  207.5000  215.0000  222.5000  230.0000  237.5000
   47.5000   60.0000   72.5000   85.0000   97.5000  110.0000  122.5000  135.0000  147.5000  160.0000  172.5000  185.0000  197.5000  210.0000  222.5000  235.0000  247.5000  260.0000  272.5000  285.0000
         0   17.5000   35.0000   52.5000   70.0000   87.5000  105.0000  122.5000  140.0000  157.5000  175.0000  192.5000  210.0000  227.5000  245.0000  262.5000  280.0000  297.5000  315.0000  332.5000
  -47.5000  -25.0000   -2.5000   20.0000   42.5000   65.0000   87.5000  110.0000  132.5000  155.0000  177.5000  200.0000  222.5000  245.0000  267.5000  290.0000  312.5000  335.0000  357.5000  380.0000
  -95.0000  -67.5000  -40.0000  -12.5000   15.0000   42.5000   70.0000   97.5000  125.0000  152.5000  180.0000  207.5000  235.0000  262.5000  290.0000  317.5000  345.0000  372.5000  400.0000  427.5000
 -142.5000 -110.0000  -77.5000  -45.0000  -12.5000   20.0000   52.5000   85.0000  117.5000  150.0000  182.5000  215.0000  247.5000  280.0000  312.5000  345.0000  377.5000  410.0000  442.5000  475.0000
 -190.0000 -152.5000 -115.0000  -77.5000  -40.0000   -2.5000   35.0000   72.5000  110.0000  147.5000  185.0000  222.5000  260.0000  297.5000  335.0000  372.5000  410.0000  447.5000  485.0000  522.5000
 -237.5000 -195.0000 -152.5000 -110.0000  -67.5000  -25.0000   17.5000   60.0000  102.5000  145.0000  187.5000  230.0000  272.5000  315.0000  357.5000  400.0000  442.5000  485.0000  527.5000  570.0000
 -285.0000 -237.5000 -190.0000 -142.5000  -95.0000  -47.5000         0   47.5000   95.0000  142.5000  190.0000  237.5000  285.0000  332.5000  380.0000  427.5000  475.0000  522.5000  570.0000  617.5000
```

Result Comparison:

```
Comparing Results of Calculation when solved without using built in functions & by using built in functions
1.1 Calculation of Trace without using built-in function Matched exactly with calculation Trace by using built-in function
1.2 Calculation of Sum, Mean & Maximum of all the entries without using built-in function Matched exactly with calculation Sum, Mean & Maximum of all the entries by using built-in function
1.3 Calculation of Sum, Mean & Maximum of all the absolute values of entries without using built-in function Matched exactly with calculation Sum, Mean & Maximum of all the absolute values of entries by using built-in function
1.4 Calculation of Row mean without using built-in function Matched exactly with calculation Row mean by using built-in function
1.5 Calculation of Column mean without using built-in function Matched exactly with calculation Column mean by using built-in function
1.6 Calculation of M * M transpose without using built-in function Matched exactly with calculation M * M transpose by using built-in function
```

Comparing Results of Calculation when solved without using built in functions & by using built in functions

1) Calculation of Trace without using built-in function Matched exactly with calculation Trace by using built-in function
2) Calculation of Sum, Mean & Maximum of all the entries without using built-in function Matched exactly with calculation Sum, Mean & Maximum of all the entries by using built-in function
3) Calculation of Sum, Mean & Maximum of all the absolute values of entries without using built-in function Matched exactly with calculation Sum, Mean & Maximum of all the absolute values of entries by using built-in function
4) Calculation of Row mean without using built-in function Matched exactly with calculation Row mean by using built-in function
5) Calculation of Column mean without using built-in function Matched exactly with calculation Column mean by using built-in function
6) Calculation of M * M transpose without using built-in function Matched exactly with calculation M * M transpose by using built-in function

2. Testing Function code 2:

N = 20, a = 0

```
Testing Function code 2; N = 20 , a = 0
Matrix A of size N × N and a Real number a where the entry A(i,j) is 1 if M(i,j) >= a and -1 otherwise.
     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1    -1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1    -1
     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
```

3.

4. **Sorting real number array into ascending order form.**

vector with combination of all real number types = [3, 6, 4, 9, 1/2, 0, 0.9, -5]

```
Given Dis-ordered Vectot A of size N x 1 = 8 x 1

    3.0000
    6.0000
    4.0000
    9.0000
    0.5000
         0
    0.9000
   -5.0000

Vectot A of size N x 1 = 8 x 1. Sorted in Ascending order

   -5.0000
         0
    0.5000
    0.9000
    3.0000
    4.0000
    6.0000
    9.0000
```

5. **<u>Computing Prime numbers up to a given maximal number using the iterative algorithm called sieve of Eratosthenes.</u>**

   Maximal number = 100

```
Sieve of Eratosthenes iterative algorithm to find Prime Numbers
It took 74 iterations to calculate all the Prime Numbers lessthan equal to 100
There are total of 25 Prime numbers less than 100, they are :
     2     3     5     7    11    13    17    19    23    29    31    37    41    43    47    53    59    61    67    71    73    79    83    89    97
```

- Sieve of Eratosthenes iterative algorithm to find Prime Numbers
- It took 74 iterations to calculate all the Prime Numbers less than equal to 100
- There are total of 25 Prime numbers less than 100, they are:
- 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97