ID:220021**108**

Assignment-05:

Exercise-01:

```
p1=input('How many test cases : ');
for j=1:p1
    Input_task01=input('Enter a Number : ')
    Output_task1=pentatope_number(Input_task01)
end
```

```
Input_task01 = 1
Output_task1 = 1
Input_task01 = 4
Output_task1 = 35
Input_task01 = 11
Output_task1 = 1001
Input_task01 = 50
Output_task1 = 292825
Input_task01 = 1×5
     1     2     3     4     5
Output_task1 = 1×5
     1     5    15    35    70
```

Exercise-02:

```
p2=input('How many test cases : ');
for j=1:p2
    Input_task02=input('Enter an array : ')
    Output_task2=lcm_array(Input_task02)
end
```

```
Input_task02 = 1×2
     6     7
Output_task2 = 42
Input_task02 = 1×2
     4    12
Output_task2 = 12
Input_task02 = 1×3
     4    12    25
Output_task2 = 300
Input_task02 = 1×9
     4    12    25     2     3    14    52    23    45
Output_task2 = 1883700
```

Exercise-03:

```
p3=input('How many test cases : ');
for j=1:p3
    Input_task03=input('Enter a Number : ')
    Output_task3= number_category(Input_task03)
end
```

```
Input_task03 = 8
Output_task3 =
```

```
'Deficient'
Input_task03 = 12
Output_task3 =
'Abundant'
Input_task03 = 8128
Output_task3 =
'Perfect'
Input_task03 = 198
Output_task3 =
'Abundant'
Input_task03 = 11088
Output_task3 =
'Abundant'
Input_task03 = 33550336
Output_task3 =
'Perfect'
```

Exercise-04:

```
p4=input('How many test cases : ');
for j=1:p4
    Input_task04_1=input('Enter an array : ')
    Input_task04_2 = input('Enter a string : ', 's');
    Output_task4=flipped_diag(Input_task04_1,Input_task04_2)
end
```

```
Input_task04_1 = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
Output_task4 = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
Input_task04_1 = 2×2
     1     5
     6     3
Output_task4 = 2×2
     1     5
     6     3
Input_task04_1 = 2×2
     1     5
     6     3
Output_task4 = 2×2
     1     5
     6     3
```

Exercise-05:

```
p5=input('How many test cases : ');
for j=1:p5
    Input_task05=input('Enter an array : ')
    Output_task5=blockStats(Input_task05)
```

```
end
```

```
Input_task05 = 1×100
   144   156    94    18    48   190    32   172   112   208    17    93    23 ···
mean_value = 1×20
   92.0000  142.8000   67.0000  123.0000  108.0000   83.4000  105.4000   56.6000 ···
max_value = 1×20
   156   208   201   181   190   181   178   107   188   197   163   199   171 ···
min_value = 1×20
    18    32     1    18    38    29    31    16    11    71    24    21    13 ···
Output_task5 = 1×20
   92.0000  142.8000   67.0000  123.0000  108.0000   83.4000  105.4000   56.6000 ···
```

Exercise-06:

```
Input_task06_1=[10, 20, 30, 40]
```

```
Input_task06_1 = 1×4
    10    20    30    40
```

```
Input_task06_2=[2, 5, 0, 10]
```

```
Input_task06_2 = 1×4
     2     5     0    10
```

```
Output_task6=vectorOps(Input_task06_1,Input_task06_2,'add')
```

```
Output_task6 = 1×4
    12    25    30    50
```

```
Output_task6=vectorOps(Input_task06_1,Input_task06_2,'divide')
```

```
Warning: Division by zero encountered. The result will contain NaN.
Output_task6 = 1×4
     5     4   NaN     4
```

Exercise-07:

```
Out=total_wind_power()
```

```
Out = 1×3    Columns 3:3
10^4 ×
    2.5158
```

Exercise-08:

```
p8=input('How many test cases : ');
for j=1:p8
    Input_task08=input('Enter an array : ')
    Output_task8=analyzeArray(Input_task08)
end
```

```
Input_task08 = 1×27
     4    -3    21    23    -5     6     0    -8     6     4    -2     4     5 ···
neg_count = 9
zero_indices = struct with fields:
    linear: [3×1 double]
       row: [3×1 double]
       col: [3×1 double]
pos_nums = 15×1
     4
    21
    23
     6
     6
     4
     4
     5
     5
    11
     ⋮
stats = struct with fields:
       mean: -3
    std_dev: 24.3942
sq_nums = 7×1
     4
     0
     4
     4
     0
     0
     9
rearranged_array = 1×27
     4     6    -8     6     4    -2     4     6     6    -6   -90    -6     8 ···
Output_task8 = 9
Input_task08 = 6×9
   228   353    52   124   217   116   400   420   260
   208   -14   397   -61   278   163   288   185   100
    -6     4   119    22     2   -75   152   170    84
    46   -27   -19   107   -57    25   203     7   434
   142     5   387   212    44   328    10   152  -102
     6   122   429    24    56  -107   135   228   376
neg_count = 9
zero_indices = struct with fields:
    linear: [1×0 double]
       row: [1×0 double]
       col: [1×0 double]
pos_nums = 1×45
   228   208    46   142     6   353     4     5   122    52   397   119   387 ···
stats = struct with fields:
       mean: 136.1481
    std_dev: 150.8051
sq_nums = 1×4
     4    25   400   100
rearranged_array = 6×9
   228   -14   212   116   420    84     5   -61    25
   208     4    24   328   170   434   397   107  -107
    -6   122   278   400   152  -102   119   217   203
    46    52     2   288   228   376   -19   -57   135
   142   124    44   152   260   353   387   163   185
     6    22    56    10   100   -27   429   -75     7
Output_task8 = 9
```