```
  ...: ## 1. Xinmiao Tan
  ...:
  ...: ## 2. Create matrix A with size (3,5) containing random numbers A = np.random.random(15)
  ...: import numpy as np
  ...: import random
  ...: random.seed(1)
  ...: A = np.random.random(15)
  ...: A.shape=(3,5)
  ...: A = np.matrix(A)
  ...: A
  ...:
Out[2]:
matrix([[0.35501806, 0.131415  , 0.09816508, 0.74440821, 0.62315052],
        [0.99945482, 0.98973921, 0.57379892, 0.8367408 , 0.73816913],
        [0.40156235, 0.76331823, 0.03868472, 0.24031897, 0.50268578]])
In[3]: ## 3. Find the size and length of matrix A
  ...: print(np.size(A))
  ...: print(A.shape)
  ...: print(len(A))
  ...:
15
(3, 5)
3
In[4]: ## 4. Resize (crop/slice) matrix A to size (3,4)
  ...: A = A[:,0:4]
  ...: print(A)
  ...:
[[0.35501806 0.131415   0.09816508 0.74440821]
 [0.99945482 0.98973921 0.57379892 0.8367408 ]
 [0.40156235 0.76331823 0.03868472 0.24031897]]
In[5]: ## 5. Find the transpose of matrix A and assign it to B
  ...: B = A.T
  ...: print(B)
  ...:
[[0.35501806 0.99945482 0.40156235]
 [0.131415   0.98973921 0.76331823]
 [0.09816508 0.57379892 0.03868472]
 [0.74440821 0.8367408  0.24031897]]
In[6]: ## 6. Find the minimum value in column 1 of matrix B
  ...: print(np.min(B[0,:]))
  ...:
0.3550180568622694
In[7]: ## 7. Find the minimum and maximum values for the en4re matrix A
  ...: print(np.min(A))
  ...: print(np.max(A))
  ...:
0.03868471649639038
0.9994548207366536
In[8]: ## 8. Create vector X (an array) with 4 random numbers
  ...: X = np.random.random(4)
  ...: print(X)
  ...:
[0.93863106 0.55637372 0.30321627 0.26262397]
In[9]: ## 9. Create a function and pass vector X and matrix A in it
  ...: def fun_9(X,A):
  ...:     print(X)
  ...:     print(A)
  ...: fun_9(X, A)
  ...:
[0.93863106 0.55637372 0.30321627 0.26262397]
[[0.35501806 0.131415   0.09816508 0.74440821]
 [0.99945482 0.98973921 0.57379892 0.8367408 ]
 [0.40156235 0.76331823 0.03868472 0.24031897]]
```

```
In[10]: ## 10. In the new function multiply vector X with matrix A and assign the result to D
   ...: def fun_10_2(X, A):
   ...:     D = X * np.array(A)
   ...:     return D
   ...: fun_10_2(X, A)
   ...: D = fun_10_2(X, A)
   ...:
   ...: def fun_10_1(X, A):
   ...:     D = X * A.T
   ...:     return D
   ...: fun_10_1(X, A)
   ...: D = fun_10_1(X, A)
   ...:
In[11]: # 11. Create a complex number Z with absolute and real parts != 0
   ...: Z = complex(np.random.random(1)+1,np.random.random(1))
   ...:
In[12]: # 12. Show its real and imaginary parts as well as it's absolute value
   ...: print(Z.real)
   ...: print(Z.imag)
   ...: print(abs(Z))
   ...:
1.9072686250921929
0.9924254703923048
2.150019051670113
In[13]: # 13. Multiply result D with the absolute value of Z and record it to C
   ...: C = D * abs(Z)
   ...: print(C)
   ...:
[[1.35797679 4.04744866 1.88438948]]
In[14]: # 14. Convert matrix B from a matrix to a string and overwrite B
   ...: B = str(B)
   ...: print(B)
   ...:
[[0.35501806 0.99945482 0.40156235]
 [0.131415   0.98973921 0.76331823]
 [0.09816508 0.57379892 0.03868472]
 [0.74440821 0.8367408  0.24031897]]
```

```
In[15]: # 15. Display a text on the screen: 'Your Name is done with HW2'
   ...: print('Xinmiao Tan is down with HW2')
Xinmiao Tan is down with HW2

In[16]:
```