1.2.

a. >> a=[1 2 3; 4 5 6];
   >> b=zeros(size(a));

   1. Create a variable “a”. Assign a 2x3 matrix to variable “a”, and the first row is 1,2,3 and the second row is 4,5,6.
   2. Create a variable “b”. Use size() to get the size of “a”, so get 2x3. Allocate a zero matrix which size is the same as “a”, 2x3. Hence, the “b” is a 2x3 zero matrix.

b. >> x=round([1.5 2; 2.2 3.1])
   >> a=find(x(:)>2);

   1. Create a variable “x”. Build a 2x2 matrix which the first row is 1.5, 2, and the second row is 2.2, 3.1. Use round() to round all elements of 2x2 matrix, and then assign the result to “x”.
   2. Make 4x1 column vector by stacking up columns of “x”, so get a column vector, [2 ; 2 ; 2 ; 3]. find() will return indices of every element which satisfies the condition, bigger than 2. Thus, we get 4. 4 is the index of “3”.

c. >> a = [1 2; 3 4; 5 6];
   >> b=a(:);
   >> c=reshape(b, 6,1)

   1.Create a variable “a” and assign a 3x2 matrix to it.
   2. Make 6x1 column vector by stacking up columns of “a”, so the vector is [1;3;5;2;4;6]. Assign it to “b”.
   3. Use elements of “b” to create a new 6x1 matrix. However, the b is already 6x1, so the return value is the same as “b”. Use c to get the return value.

d. >>a=rand(1, 100);
   >>b=randperm(100);
   >>c=a(b(1:5));

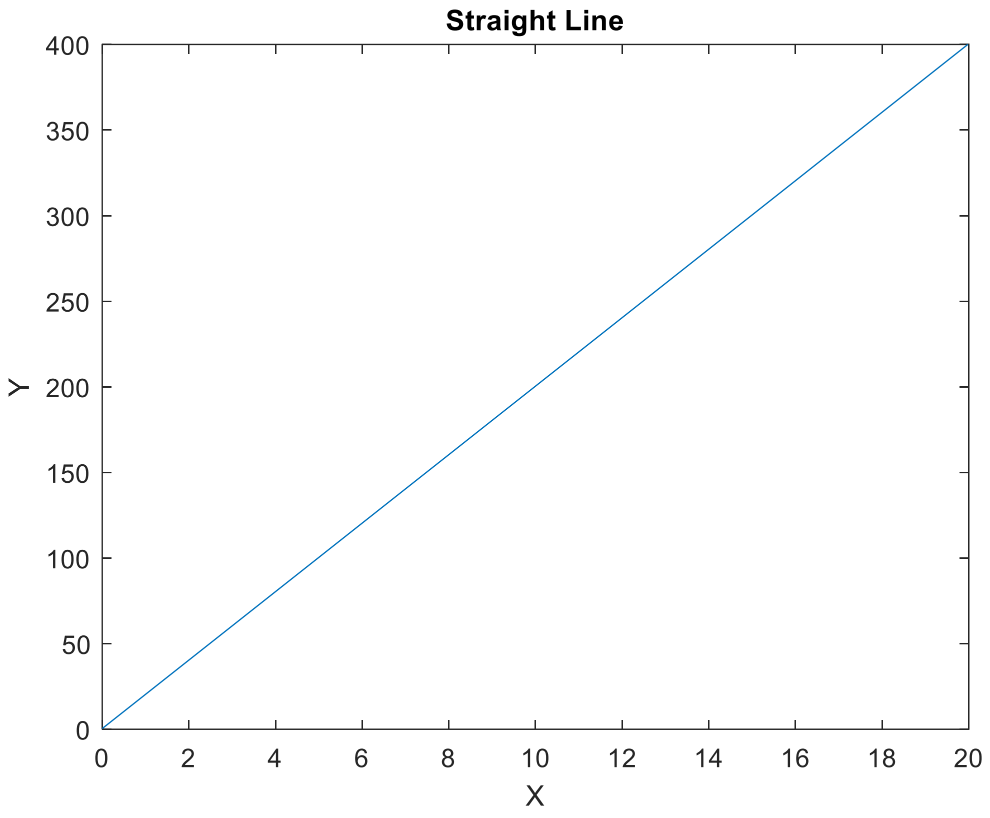
   1. Randomly filled 1x100 matrix and assign it to “a”.
   2. Return a row vector which randomly fills permutation of the integers from 1 to 100.
   3.Firstly, get the elements of “b” by indices from 1 to 5. Secondly Get elements of “a” by the indices of first step. Assign the final result to “c”.

e. >>a=[100:200];
   >>b=find(a>120);
   >>c=a(b);

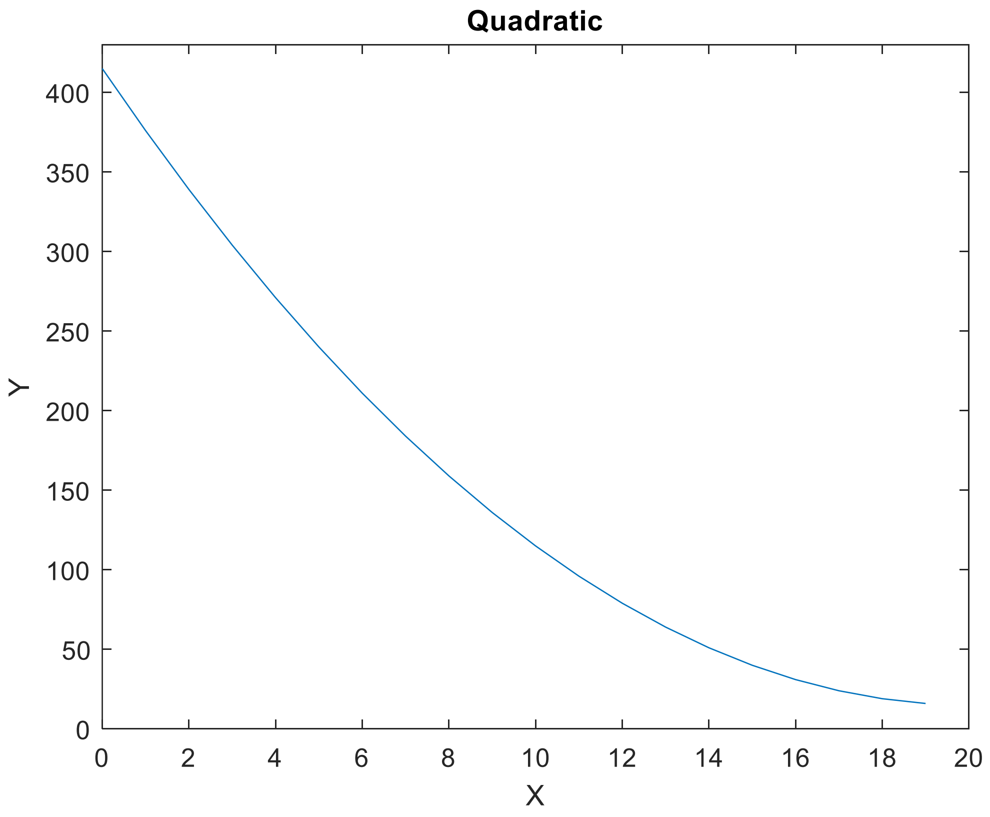
   1. Create a 1x101 row vector from 100 to 200, so get [100 101 102 … 200]. Assign the row vector to “a”.
   2. find() will return indices of every element which satisfies the condition, bigger than 120, and assign to “b”. Hence, “b” is 1x80 row vector, [22 23 24 … 101].
   3. Get the elements of “a” by the indices which “b” contains. Assign the list of elements to “c”. Hence, “c” is a 1x80 row vector, [121 122 123 … 200].

1.3.

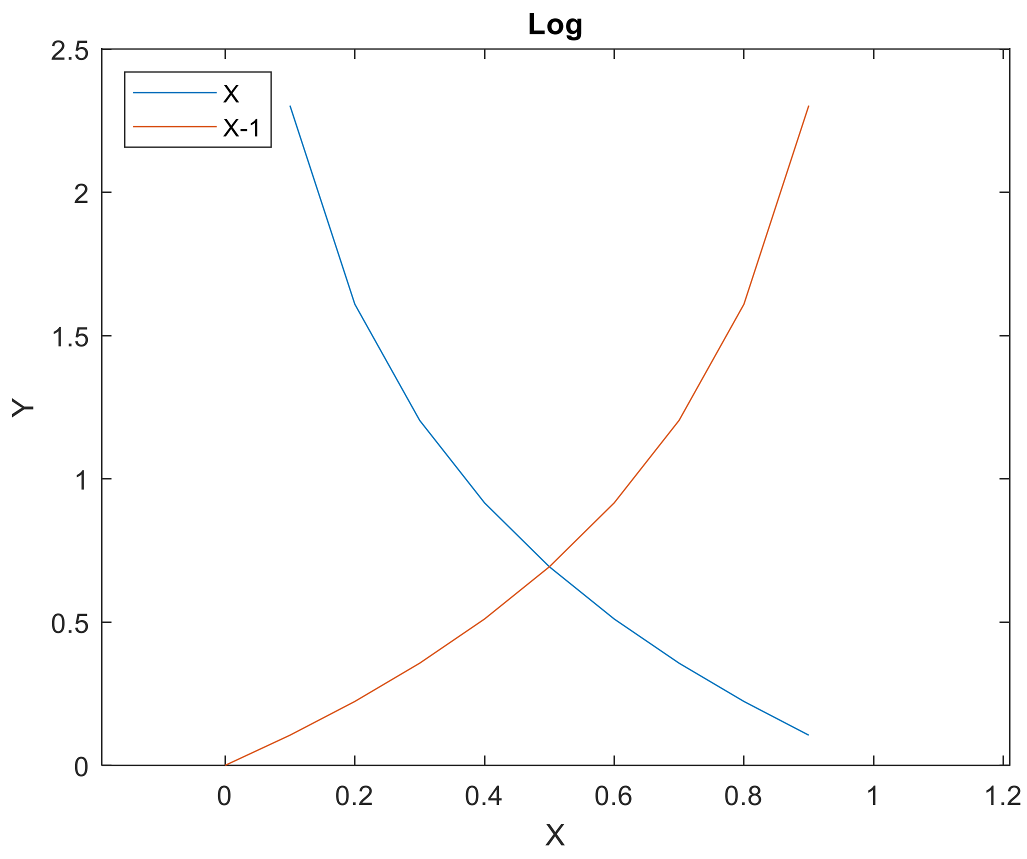
- straight line $y = \theta_0 + \theta_1 x$ where $\theta_0 = 20,\ \theta_1 = 0.4$



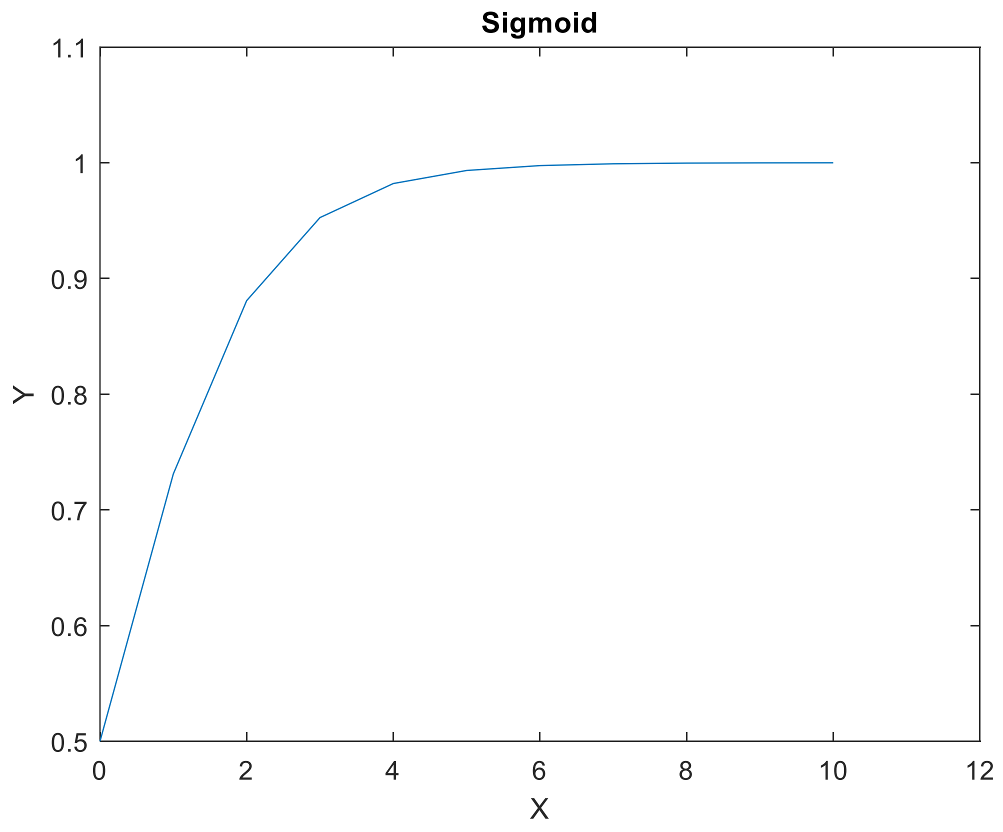
- quadratic function: $y = (x - \theta_1)^2 + \theta_0, where\ \theta_1 = 20, \theta_0 = 15$



- log function, $y = -\log(x)$ and $y = -\log(1 - x)$



- sigmoid function, $y = 1/(1 + e^{-x})$

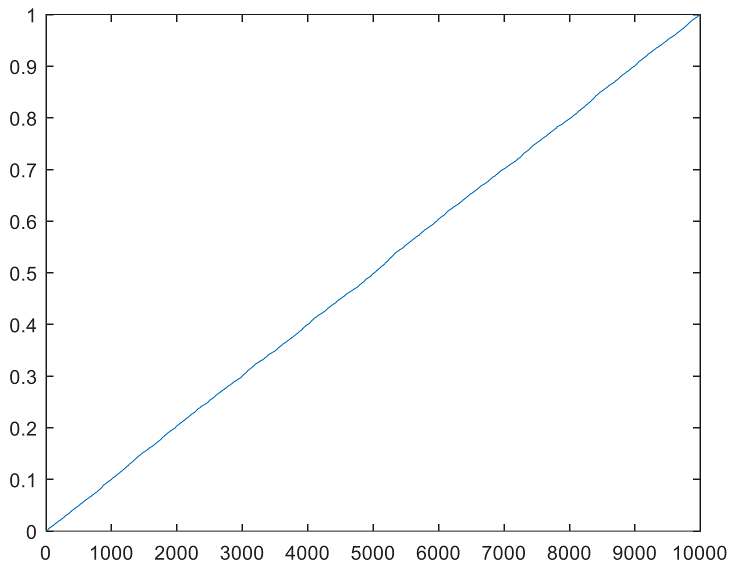


1.4. Given a matrix A=rand(100,100).

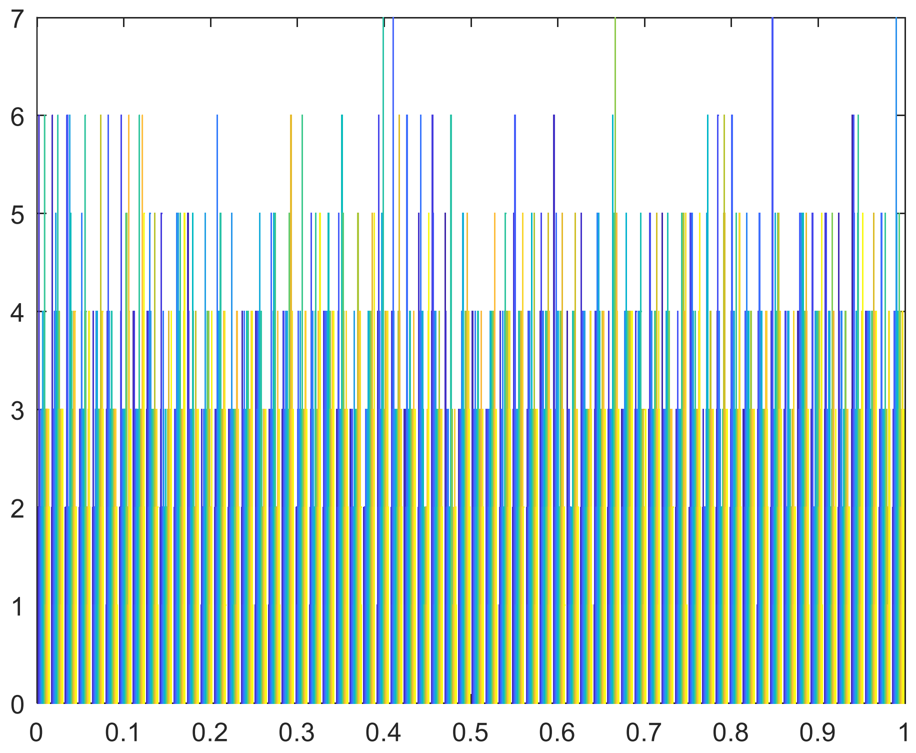
a. Sort all the elements in A, put the result in a single 10,000-dimensional vector x, and plot the values in x.

Ans:

```
A = rand(100,100);
x = sort(A(:));
x = reshape(x, 1, 10000);
plot(x);
```



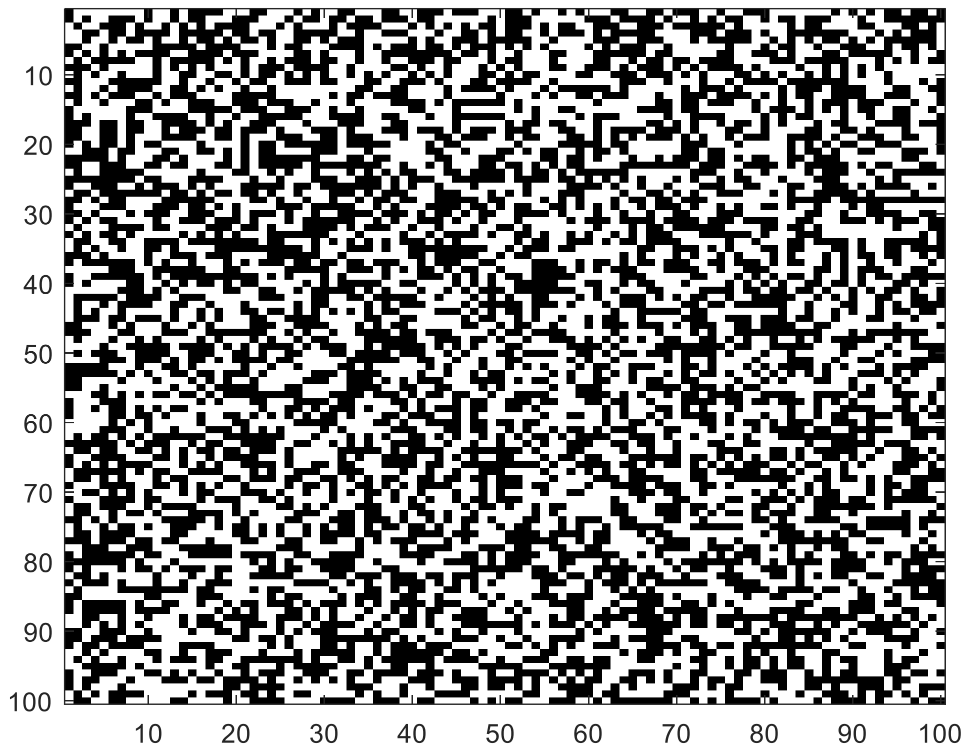
b. Create a 64-bin histogram bar chart of the elements in A. Use hist();
Ans: `hist(A,64);`



c. Create a new matrix with the same size as A, which is 255 wherever the element in A is greater than a threshold t (e.g., 0.5), and 0 everywhere else; call imagesc() to visualize this matrix as an grayscale image;

<u>Ans:</u>

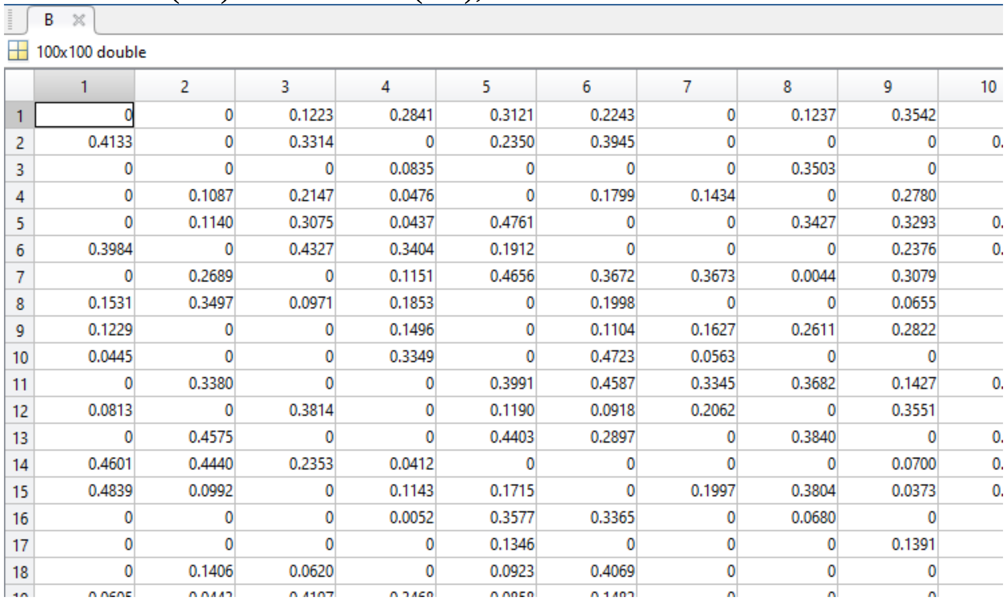
```
ind = find (A > 0.5);
threshold = zeros(size(A));
TwoFiveFive = 255*ones(size(A));
threshold(ind) = TwoFiveFive(ind);
imagesc(threshold);
colormap gray;
```



d. Create a matrix to store the elements in the bottom right quadrant of A.
Ans: BotRight = A(end/2+1:end, end/2+1:end);

e. Create a new matrix that is a duplicate of A; Subtract A's mean value from every element of B; Set any negative element in B to be 0.

Ans:
```
B = A;
meanA = mean(A(:));
B = B - meanA;
ind = find(B <0);
ZeroMatrix = zeros(size(B));
B(ind) = ZeroMatrix(ind);
```



	1	2	3	4	5	6	7	8	9	10
1	0	0	0.1223	0.2841	0.3121	0.2243	0	0.1237	0.3542	
2	0.4133	0	0.3314	0	0.2350	0.3945	0	0	0	0.
3	0	0	0	0.0835	0	0	0	0.3503	0	
4	0	0.1087	0.2147	0.0476	0	0.1799	0.1434	0	0.2780	
5	0	0.1140	0.3075	0.0437	0.4761	0	0	0.3427	0.3293	0.
6	0.3984	0	0.4327	0.3404	0.1912	0	0	0	0.2376	0.
7	0	0.2689	0	0.1151	0.4656	0.3672	0.3673	0.0044	0.3079	
8	0.1531	0.3497	0.0971	0.1853	0	0.1998	0	0	0.0655	
9	0.1229	0	0	0.1496	0	0.1104	0.1627	0.2611	0.2822	
10	0.0445	0	0	0.3349	0	0.4723	0.0563	0	0	
11	0	0.3380	0	0	0.3991	0.4587	0.3345	0.3682	0.1427	0.
12	0.0813	0	0.3814	0	0.1190	0.0918	0.2062	0	0.3551	
13	0	0.4575	0	0	0.4403	0.2897	0	0.3840	0	0.
14	0.4601	0.4440	0.2353	0.0412	0	0	0	0	0.0700	0.
15	0.4839	0.0992	0	0.1143	0.1715	0	0.1997	0.3804	0.0373	0.
16	0	0	0	0.0052	0.3577	0.3365	0	0.0680	0	
17	0	0	0	0	0.1346	0	0	0	0.1391	
18	0	0.1406	0.0620	0	0.0923	0.4069	0	0	0	

f. Create a new matrix to include all rows of A whose first column is larger than 0.5 and second column is smaller than 0.8;

Ans:
```
ind = A(:,1) > 0.5 & A(:,2) < 0.8;
fmatrix = A(ind,:);
```

g. Create a new matrix by randomly selecting 20 rows from A.

Ans:
```
ind = randi(100,[1 20]);
RandMatrix = A(ind,:);
```

1.5 Use the function rand () to write a function that returns the roll of a six-sided die.

Ans:
```
function y = rollsixsidedie()
    y=ceil(6*rand());
```