**2. Describe (in words where appropriate) the result of each of the following Matlab commands. Use the help  
command as needed, but try to determine the output without entering the commands into Matlab. Do not submit a screenshot of the result of typing these commands.**

**a. >> x = randperm (1000);**

Function: After execution of this command, x will contain a row matrix containing the random permutations of the integers in range [1, 1000].

**b. >> a = [1, 2, 3; 4 5 6; 7 8 9] ;  
>> b = a (2, :);**

Function: The first statement will define a 3x3 matrix containing {{1 2 3}, {4 5 6}, {7 8 9}} in the variable a. The second statement, on execution, will return the entire second row of the matrix a in the variable b.

**c. >> f = randn (5,1);  
>> g = f(find(f > 0));**

Function: Execution of first command will return a column matrix with 5 entries in variable f containing random values. The execution of the second command will find and make a matrix of the only positive values in f and will store it into g which will also be a column matrix.

**d. >> x = zeros (1, 10) + 0.5;   
>> y = 0.5.\*ones (1, length(x)) ;  
>> z = x + y;**

Function: The first line on execution will make a matrix of order 1x10 and initialize all values to 0. After that it will add 0.5 to each value and the resultant matrix will be stored in x.

The second line will make a matrix of size 1 x (length of matrix x, that is 10 according to the previous statement) and with all entries equal to one. Later it will multiply every entry with 0.5 and the resultant matrix will be stored in y.

The last line will add these two 1x10 matrices and will store the resultant matrix in variable z. That will have all the values equal to 1.

**e. >> a = [1:100];   
>> b = a ([end: ‐1:1]);**

Function:

The first line will make a matrix of order 1x100. The variable a will act as a counter being a 1x1 matrix. The values of the counter will be assigned at every index. It will contain [1 2 3 . . . 100].

The second line upon execution will traverse the matrix ‘a’ in reverse order (from end index to the first index ‘1’ with decrement of ‘-1’) while assigning all the values in a to b at the corresponding indices. The matrix b will be of the order 1x100. It will contain [100 99 98 . . . 1].