lab02_HW

Work at ubuntu16.04 + Matlab R0217a academic use

1. Identify what is wrong with the following MATLAB command sequences and/or answers.

   (i)

   ```
   >> x=0:10;
   >> y+1=x;
   ```

   Wrong
   Assignment should contain only one variable

   ```
   >> y = x + 1
   y =
    1     2     3     4     5     6     7     8     9    10    11
   ```

   (ii)

   ```
   >> (2+7)\15+12
   ans = 0.3333
   ```

   Wrong
   This means 15/9 + 12

   ```
   ans = 13.6667
   ```

   (iii)

   ```
   >> B=0:pi/2:5*pi;
   >> B(0)
   ans = 0
   ```

   Wrong
   Matlab index is start at 1

```
>> B(1)
ans = 0
```

(iv)

```
>> c=[5 8 7 ; 4 5 8]
c = 5 8 7; 4 5 8
```

Wrong. it should be matrix

```
c =

 5     8     7
 4     5     8
```

(v)

```
>> A=[4,8,1]; B=[5,9,6];
>> A*B
```

Wrong. Dimension error
It should be same dimentsion

```
>> A * B'
ans = 98
```

2. Make the following variables:
   (1) `a = [5 4.8 4.6 ⋯ −4.8 −5]`
   `>> a = 5:-0.2:-5`
   (2) `b = [10^0 10^0.01 10^0.02 ⋯ 10^0.99 10^1 ]`
   `>> 10.^[0:0.01:1]`
   (3) `c = Hello`
   `>> c = "Hello"`

3. Make the following matrices and calculate their eigenvalues and eigenvectors:

$$A = \left[\begin{pmatrix} 2 & \cdots & 2 \\ \vdots & \ddots & \vdots \\ 2 & \cdots & 2 \end{pmatrix}\right] \in \Re^{9\times 9}$$

```
>> A = ones(9, 9) * 2
>> [vect val] = eig(A) % eigenvector and eigenvalue

vect =

   -0.8616   -0.0000   -0.0000         0         0    0.0000    0.2924   -0.2472    0
    0.0922   -0.1826   -0.2236    0.2184   -0.7071    0.4498   -0.1927   -0.0997    0
    0.0922   -0.1826   -0.2236    0.4176    0.0000   -0.7587   -0.1927   -0.0997    0
    0.0922   -0.1826   -0.2236   -0.8545   -0.0000   -0.1409   -0.1927   -0.0997    0
    0.0922   -0.1826    0.8944    0.0000   -0.0000    0.0000   -0.1927   -0.0997    0
    0.0922    0.9129    0.0000         0   -0.0000    0.0000   -0.1927   -0.0997    0
    0.0922   -0.1826   -0.2236    0.2184    0.7071    0.4498   -0.1927   -0.0997    0
    0.4363   -0.0000   -0.0000         0         0    0.0000    0.8311   -0.0881    0
   -0.1279    0.0000         0         0         0         0    0.0324    0.9335    0


val =

   -0.0000         0         0         0         0         0         0         0
         0   -0.0000         0         0         0         0         0         0
         0         0   -0.0000         0         0         0         0         0
         0         0         0   -0.0000         0         0         0         0
         0         0         0         0   -0.0000         0         0         0
         0         0         0         0         0    0.0000         0         0
         0         0         0         0         0         0    0.0000         0
         0         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0         0   18
```

$$B = \begin{bmatrix} 1 & 11 & \cdots & 91 \\ 2 & 12 & \ddots & 92 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 20 & \cdots & 100 \end{bmatrix}$$

```
>> B = reshape(1:100, 10, 10)
>> [vect val] = eig(B) % eigenvector and eigenvalue

vect =

  Columns 1 through 6

  -0.2885 + 0.0000i  -0.5865 + 0.0000i  -0.7920 + 0.0000i  -0.6304 + 0.0000i  -0.085
  -0.2945 + 0.0000i  -0.4901 + 0.0000i   0.2928 + 0.0000i   0.4149 + 0.0000i   0.158
  -0.3006 + 0.0000i  -0.3937 + 0.0000i   0.2424 + 0.0000i   0.3468 + 0.0000i  -0.017
  -0.3067 + 0.0000i  -0.2973 + 0.0000i   0.2642 + 0.0000i   0.0929 + 0.0000i  -0.032
  -0.3127 + 0.0000i  -0.2009 + 0.0000i   0.3055 + 0.0000i  -0.3416 + 0.0000i   0.435
  -0.3188 + 0.0000i  -0.1045 + 0.0000i   0.0349 + 0.0000i   0.2424 + 0.0000i  -0.586
  -0.3248 + 0.0000i  -0.0081 + 0.0000i   0.0289 + 0.0000i   0.0228 + 0.0000i   0.055
  -0.3309 + 0.0000i   0.0883 + 0.0000i  -0.0982 + 0.0000i   0.1740 + 0.0000i  -0.202
  -0.3370 + 0.0000i   0.1847 + 0.0000i  -0.0535 + 0.0000i  -0.3085 + 0.0000i   0.219
  -0.3430 + 0.0000i   0.2811 + 0.0000i  -0.2250 + 0.0000i  -0.0133 + 0.0000i   0.054

  Columns 7 through 10

   0.5316 + 0.0000i   0.5316 + 0.0000i  -0.3036 + 0.0000i  -0.2627 + 0.0000i
  -0.0876 - 0.0137i  -0.0876 + 0.0137i   0.1573 + 0.0000i   0.1581 + 0.0000i
  -0.4564 - 0.0156i  -0.4564 + 0.0156i   0.2232 + 0.0000i   0.2370 + 0.0000i
  -0.2302 - 0.0567i  -0.2302 + 0.0567i   0.0354 + 0.0000i   0.1344 + 0.0000i
  -0.0793 + 0.3087i  -0.0793 - 0.3087i   0.4896 + 0.0000i  -0.3922 + 0.0000i
   0.3347 - 0.2176i   0.3347 + 0.2176i  -0.6628 + 0.0000i  -0.3024 + 0.0000i
  -0.1749 + 0.1585i  -0.1749 - 0.1585i  -0.1905 + 0.0000i   0.2060 + 0.0000i
  -0.0324 - 0.2385i  -0.0324 + 0.2385i   0.0862 + 0.0000i   0.4547 + 0.0000i
   0.1389 - 0.1125i   0.1389 + 0.1125i   0.3013 + 0.0000i   0.2773 + 0.0000i
   0.0555 + 0.1874i   0.0555 - 0.1874i  -0.1361 + 0.0000i  -0.5102 + 0.0000i


val =

   1.0e+02 *

  Columns 1 through 6

   5.2084 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i  -0.1584 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.000
```

```
  Columns 7 through 10

   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
```

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>> C = diag([1:5, 4:-1:1])
>> [vect val] = eig(C) % eigenvector and eigenvalue
vect =

     1     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     1     0     0
     0     0     0     0     1     0     0     0     0
     0     0     1     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0


val =

     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0
     0     0     0     0     3     0     0     0     0
     0     0     0     0     0     3     0     0     0
     0     0     0     0     0     0     4     0     0
     0     0     0     0     0     0     0     4     0
     0     0     0     0     0     0     0     0     5
```

4. An M-by-M matrix X is given. Without using loops, extract values from matrix X to create the following:
   Let's set

```
>> n = 9;
>> X = reshape(1:n^2, n, n) % use this to check answer
X =
 1    10    19    28    37    46    55    64    73
 2    11    20    29    38    47    56    65    74
 3    12    21    30    39    48    57    66    75
 4    13    22    31    40    49    58    67    76
 5    14    23    32    41    50    59    68    77
 6    15    24    33    42    51    60    69    78
 7    16    25    34    43    52    61    70    79
 8    17    26    35    44    53    62    71    80
 9    18    27    36    45    54    63    72    81
```

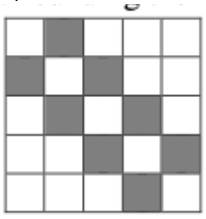
   i) matrix A – composed of all values in odd columns AND odd rows

```
>> X(1:2:n, 1:2:n)
ans =
 1    19    37    55    73
 3    21    39    57    75
 5    23    41    59    77
 7    25    43    61    79
 9    27    45    63    81
```

ii) matrix B – composed of all entries of X, except for the outside rows and columns

```
>> X(2:n-1, 2:n-1)
ans =
11    20    29    38    47    56    65
12    21    30    39    48    57    66
13    22    31    40    49    58    67
14    23    32    41    50    59    68
15    24    33    42    51    60    69
16    25    34    43    52    61    70
17    26    35    44    53    62    71
```

iii) matrix C – composed of diagonals surrounding the middle diagonal of matrix X



```
>> X .* (diag(ones(1, n - 1), 1) + diag(ones(1, n - 1), -1))
ans =
 0    10     0     0     0     0     0     0     0
 2     0    20     0     0     0     0     0     0
 0    12     0    30     0     0     0     0     0
 0     0    22     0    40     0     0     0     0
 0     0     0    32     0    50     0     0     0
 0     0     0     0    42     0    60     0     0
 0     0     0     0     0    52     0    70     0
 0     0     0     0     0     0    62     0    80
 0     0     0     0     0     0     0    72     0
```

5. Without using loops, calculate the sum of the following series
   1 - 1/2 + 1/3 - 1/4 + 1/5 …
   for the first 10,000 terms.
   We can do it for only one line

```
>> sum([arrayfun(@(x) 1 / x - 1/(x + 1), [1:2:10000])])
ans =
0.6931
```

The answer is same as `ln(2)` in Math

6. (1) Load the image 02Lena.bmp by typing:

```
A = imread('02Lena.png');
```

What is the type of variable A?

```
>> class(A) % get the type
ans =
'uint8'
```

so it is a matrix of uint8

(2) Display the image by typing:

```
imshow(A);
```

Now multiply the entries of A to 1.5. Display the image and report what you observe.

```
>> subplot(1, 2, 1);
>> title("Original");
>> imshow(A);
>> title("Original");
>> subplot(1, 2, 2);
>> imshow(A * 1.5);
>> title("Original * 1.5");
```

And you will notice that multiply 1.5 will be brighter


Original


Original * 1.5