

I. Using Matlab [60 points]

Talk with Qiyang Tuo

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.

[10 points]

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

`//This will generate a random permutation of the integers from 1 to 1000`

`// x=[2 4 1 1000]`

b. `>> a = [1,2,3; 4 5 6; 7 8 9];`

`>> b = a(2,:);`

`// b will get the second row of a`

`// b = [4,5,6]`

c. `>> a = [1,2,3; 4 5 6; 7 8 9];`

`>> b = a(:);`

`//This will build a new nx1 matrix b`

`// b=[1;2;3;4;5;6;7;8;9]`

d. `>> f = randn(5,1);`

`>> g = f(find(f > 0));`

`// randn will generate a 5x1 matrix containing pseudorandom values drawn from the standard normal distribution(from help matlab)`

`// g= the positive number find from f.`

`// eg: f= [0.6;0.5;-2;0.4;0.2]`

`// g=[0.6;0.5;0.4;0.2]`

e. `>> x = zeros(1,10)+0.5;`

```
>> y = 0.5.*ones(1,length(x));

>> z = x + y;
//x=10 zero's + 0.5
//x=[0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5];
// the same length with x, 10 one's times 0.5
//y=[0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5];
//z=[1,1,1,1,1,1,1,1,1,1];
```

```
f. >> a = [1:100];
```

```
>> b = a([end:-1:1]);

// 100 x1 matrix with value from 1 to 100;
//a=[1,2,3,4.....100];
// b = invert the order of a
//b=[100,99,98,97.....1];
```

3. Write a few lines of code to do each of the following. Copy and paste your code into the

answer sheet. [20 points]

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

```
// creat roll.m
// in roll.m

//function r=roll()
//r= floor(1+(6)*rand(1,1));
```

b. Let y be the vector: $y = [1 \ 2 \ 3 \ 4 \ 5 \ 6]'$. Use the reshape command to form

a new matrix Z that looks like this: $Z = [$

1 3 5

2 4 6

$]$

```
//x=y';
//z=reshape(x,2,3);
```

c. Use the max and find functions to set x to the maximum value that occurs in Z

(above), and set r to the row it occurs in and c to the column it occurs in.

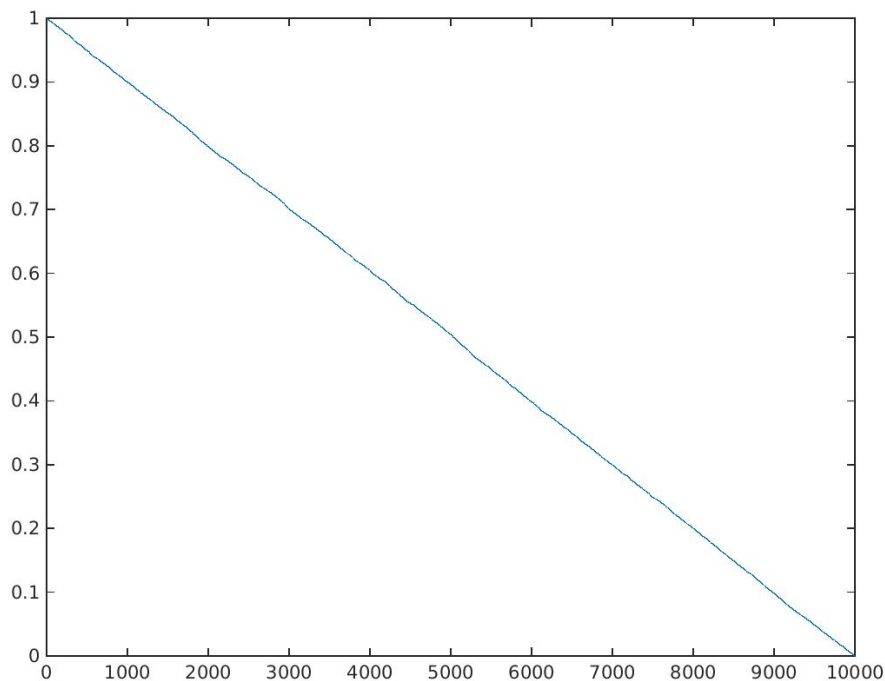
```
//x=max(max(z))  
//[r,c]=find(z==max(max(z)))
```

d. Let v be the vector: $v = [1 \ 8 \ 8 \ 2 \ 1 \ 3 \ 9 \ 8]$. Set a new variable x to be the number of 1's in the vector v .

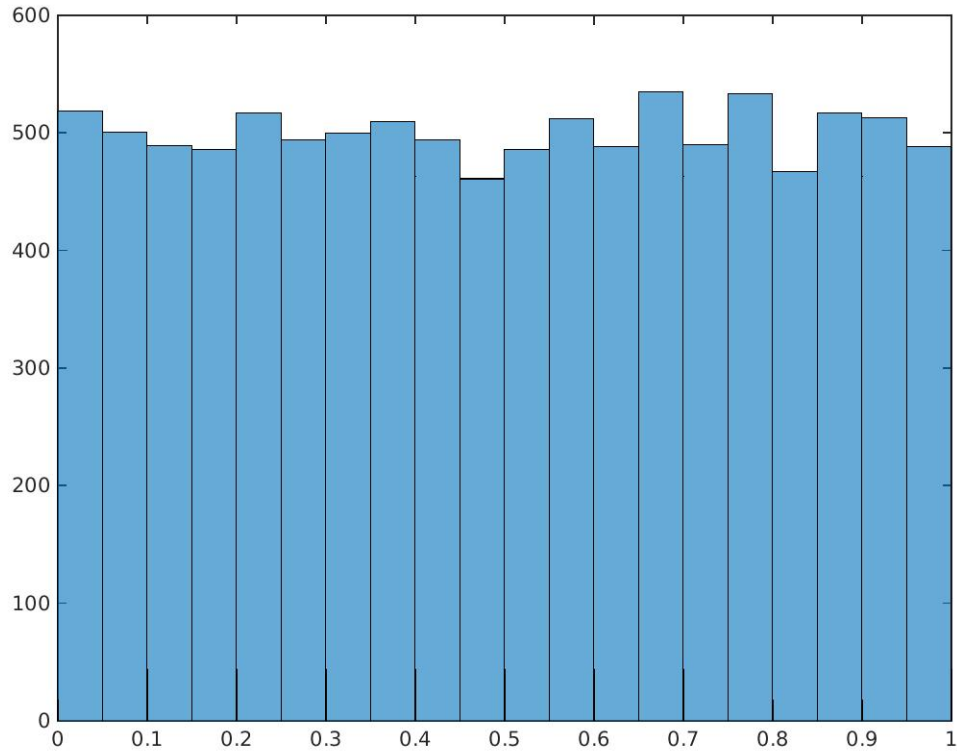
```
// a=find(1==v)  
// x=size(a,2)
```

4. Create any 100×100 matrix A (not all constant). Save A in a .mat file called PS0_A.mat and submit it. Write a script which loads PS0_A.mat and performs each of the following actions on A . Name it PS0_Q1.m and submit it. Try to avoid using for loops. [30 points]

a. Plot all the intensities in A , sorted in decreasing value. Provide the plot in your answer sheet. (Note, in this case we don't care about the 2D structure of A , we only want to sort all the intensities in one list.)



b. Display a histogram of A's intensities with 20 bins. Provide the plot in your answer sheet.



c. Create a new matrix Z that consists of the bottom left quadrant of A. Display Z as an image in your answer sheet using imshow.



d. Generate a new image W, which is the same as A, but with A's mean intensity value subtracted from each pixel. Display W as an image in your answer sheet using imshow.



e. Create and display a new matrix Y that represents a color image with the same size as A, but with 3 channels to represent R G and B values. Set the values of Y to be green (i.e., $R = 0$, $G = 255$, $B = 0$) wherever the intensity in A is greater than a threshold $t = \text{the mean intensity of A}$, and black (i.e., $R = 0$, $G = 0$, $B = 0$) everywhere else. Plot Y using `imagesc`. Look at the documentation for `colormap`. Try `colormap gray`, `colormap jet`, and others. Provide two plots (each using a different colormap) in your answer sheet.



II. Short programming example [40 points]

Write functions to do each of the following to an input color image, and then write a script that loads an image, applies each transformation to the original image, and displays the results in a figure using the Matlab subplot function in a 3x2 grid (3 rows and 2 columns). Label each subplot with an appropriate title. Name the script PS0_Q2.m.

Apply the script to a color image of your choosing, and show the results.

- Convert the input color image into a grayscale image.
- Convert the grayscale image to its “negative image”, in which the lightest values appear dark and vice versa (i.e., 0 is mapped to 255, 255 is mapped to 0, etc.)
- Map the input color image to its “mirror image”, i.e., flipping it left to right.
- Swap the red and green color channels of the input color image.
- Average the input color image with its mirror image (use typecasting!)
- Add or subtract a random value between [0,255] to every pixel in the grayscale image, then clip the resulting image to have a minimum value of 0 and a maximum value of 255.

grayscale



negativegray



mirror



swap



avg



subFromGray

