] Write a Matlab function

a- check if image is Color or not

```
function checkColor = check(I)
   I = imread('M.Png');
   if  size(I,3) == 3
       disp('Image is Color');
   else if  size(I,3) == 1
       disp('Image is not Color');
   end
end
```

b- check if image is bright or not   200 Consider bright

```
function  checkbright = check(R)
    R = imread('M.Png');
    I = rgb2gray (R);
    J = im hist (I);
    K = J> 200;
    L = J<= 200;
    if  K>L
        disp ('bright');
    else
        disp ('dark');
    end
end
```

Note
Play value 200, 210, etc.

c- check if image is binary or not

```
function  checkbinary = check(X)
    i = imread('M.Png');
    [rcd] = size(i);          j=rgb2gray(i);
    R= i(:,:,1);      G = i(:,:,2);      B=i(:,:,3);
    imshow (R);
end
```

Page: Date:

d- Called add(I,v) take two argument I as image v as value. That add value v to all pixels of image

```
function Add = add(I,v)
    B = unit8 (double(I) + v);
    disp (B);
end
```

E- Convert RGB image to gray image

```
function grayImage = grayscale (rgbimage)
    im = imread ('m.png');
    Gim = unit8 (zeros(size(Im,1), size(Im,2)));
    for i=1 : size(im,1)
        for j=1 : size(Im,2)
            Gim(i,j) = 1/3 * [im(i,j,1) + im(i,j,2) + im(i,j,3)];
        end    Note 1/9 Min or max (im(i,j,1), im(i,j,2), im(i,j,3))
    end
    figure, imshow (im), figure ; imshow (Gim) ;
end
```

f- outlier Method

```
function res = outlier (im,d)
    f = ones (3,3) /8 ;
    imd = im2double (im);
    imf = filter2 (f, imd);
    r = abs (imd-imf)-d >0 ;
    res = im2unit8 (r.*imf + (1-r).*imd);
    imshow (res);
end
```

أشرف بلال

g- Apply Median filter on image

Function Apply noise

```
function Median = Median filter (X)
        R = imread ('M.Png');        I = rgb2gray (R);
        J = imnoise ( R, 'salt & Peper', 0.2);
        K = medfilt2 (J);
        figure, imshow (J), figure, imshow (k);
end
```

h- Apply avarage filter on image

```
function avarage = avarage filter (x)
        R = imread ('M.Png');
        J = imnoise ( R, 'gaussian' );
        K = fspecial ('avarage');
        L = filter2 (K, J);
        figure, imshow(L); or disp (L);
end
```

i- Converts the gray scale image to a binary image

```
function grayscale = binaryimage (X)
        im = imread ('M.Png');
        gray = rgb2gray (im);
        adj-im = imadjust ( gray, [0.3, 0.7], [ ]);
        bw-im = im2bw (adj-im);
        figure, imshow (adj-im), figure, imshow( bw-im);
end
```

1 Return Max number of Array

```
function max = array (a)
    Max = array (1);
    for i = 1 : length (array)
        if array (i) > Max
            Max = Array(a)
        end
    end
    disp ('The max is:') Max
end
```

2 Check number Prime or not

```
function result = MyPrime (n)
        result = 0;
        for i = 1 : 1 : n
            if ( Mod(n,i) == 0)
                result = result + 1;
            end
        end
        if (result == 2)   result = 'Prime';
        else    result = 'not Prime';
end
```

- Code: Negitive image

```
function y = Myfun3(I)
 [n M] = size(I);
 m = Max( max(I)) ;
 if M>100
       X = 255 ;
  else       X=1 ;
 for i=1 :n
    for j=1 : M
         Y(i,j) = X-I(i,j) ;
    end
  end end
end
```

- Convert image to binary

```
function Y = myfun(I,T)
   [n M] = size(I);
   for i=1 :n
      for j=1:M
          if I(i,j) > T
             Y(i,j) = 1 ;
          else
             Y(i,j) = 0 ;
          end
      end
    end
end
```

- Convert RGB to grayscale

```
function y = myfun4(I)
[n M C] = size(I);
 for i=1 :n
    for j=1:M
       L = [I(i,j,1), I(i,j,2) , I(i,j,3)]
       Max Value = max(L)
       Min Value = Min(L)
   Y(i,j) = (MaxValue+minValue);
    end
  end
end
```

or

```
function y = myfun2(I)
 [n M C] = size(I);
  for i=1 :n
     for j=1 :M
  Y(i,j) = (I(i,j,1) + I(i,j,2)
            +I(i,j,3)) / 3 ;
     end
   end
end
```