

به نام خدا

نامی نذیری – گزارش تکلیف شماره 1

سوال 1:

این سوال با فرض اینکه تصویر داده شده در هدر اش کامنتی ندارد وارد برنامه شده است.

هدر را با توجه به اسپیس های موجود جداسازی کرده (4 اسپیس داریم که می شود عدد جادویی، ارتفاع، سطر و بزرگترین عدد)

```
function [output,width,height]= LoadImage(path)
    F = fopen(path);
    ImageRawData = fread(F,inf,'uint8=>char');

    %Finding white spaces indices
    Header = find(isspace(ImageRawData),4);

    %calculate width and height of the image
    height = str2num(ImageRawData(uint8(Header(1)):uint8(Header(2))));
    width = str2num(ImageRawData(uint8(Header(2)):uint8(Header(3))));

    %Separate data from headers
    ImageData = ImageRawData(Header(4) + 1:end);

    % Create RGB array of image
    RGBImageData = zeros( width, height, 3);

    % Fill the RGB array
    counter = 1;
    for i=1 : width
        for j=1 : height
            for k = 1 : 3
                RGBImageData(i,j,k) = ImageData(counter);
                counter = counter + 1;
            end
        end
    end
    output = uint8(RGBImageData);
end
```

سوال:2

```
function output = SpecialEffect(im,width,height)

width = size(im,1);
height = size(im,2);

% create gray image
GrayImage = rgb2gray(im)

% properties of the oval
OvalHalfWidth = 200;
OvalHalfHeight = 400;

OvalHalfWidthPow2 = OvalHalfWidth * OvalHalfWidth;
OvalHalfHeightPow2 = OvalHalfHeight * OvalHalfHeight ;

%Properties of the dimond
DimondHalfWidth = 100;
DimondHalfHeight = 200;

for i=1 : width
    for j=1 : height
        % Check if the pixels is insinde the Oval and outside of the dimond
        if (round((((i - (width/2))^2) / OvalHalfHeightPow2) + (((j - (height/2))^2) / OvalHalfWidthPow2)
            im(i,j,1) = 255;
            im(i,j,2) = 0;
            im(i,j,3) = 0;
        end
        % If its outside of the oval turn the pixel into gray
        if (round((((i - (width/2))^2) / OvalHalfHeightPow2) + (((j - (height/2))^2) / OvalHalfWidthPow2)
            im(i,j,1) = GrayImage(i,j);
            im(i,j,2) = GrayImage(i,j);
            im(i,j,3) = GrayImage(i,j);
        end
    end
end

% write the image into jpg file
imwrite(im,'Q2.jpg','Quality',100);
output = uint8(im);

end
```

سوال:3

```
function output = CustomRotate(im, angle)

% Calculate width and height of the image
width = size(im,1);
height = size(im,2);

% Create rotation matrix
c = cos(deg2rad(angle));
s = sin(deg2rad(angle));

R = [c -s;s c];
```

```

% Calculate bounding box of the rotated recangle (picture)
corners = [[0,0,width,width];[0,height,0,height]];
minX = 0; minY= 0;
maxX= 0; maxY= 0;
for i=1 : 4
    vec = int16(round(R * corners(:,i)));

    if( vec(1) < minX)
        minX = vec(1);
    end

    if( vec(1) > maxX)
        maxX = vec(1);
    end

    if( vec(2) < minY)
        minY = vec(2);
    end

    if( vec(2) > maxY)
        maxY = vec(2);
    end

end

% New height and width based on the bounding box
newHeight = maxY - minY;
newWidth = maxX - minX;

% buffer for the rotated image
NewRotatedImage = zeros( newWidth, newHeight, 3);

% Create matrix of the negate of the angle
c = cos(deg2rad(-angle));
s = sin(deg2rad(-angle));

R = [c -s;s c];

for i=1 : double(newWidth)
    for j=1 : double(newHeight)
        NewPos = round( R * transpose([double(i + minX),double(j +minY)]));
        NewPos = transpose(int16(NewPos));
        if(NewPos(1) > 0 && NewPos(2)>0 && NewPos(1)<width && NewPos(2)<height)
            NewRotatedImage(i,j,1) = im(NewPos(1) + 1,NewPos(2) + 1,1);
            NewRotatedImage(i,j,2) = im(NewPos(1) + 1,NewPos(2) + 1,2);
            NewRotatedImage(i,j,3) = im(NewPos(1) + 1,NewPos(2) + 1,3);
        end
    end
end

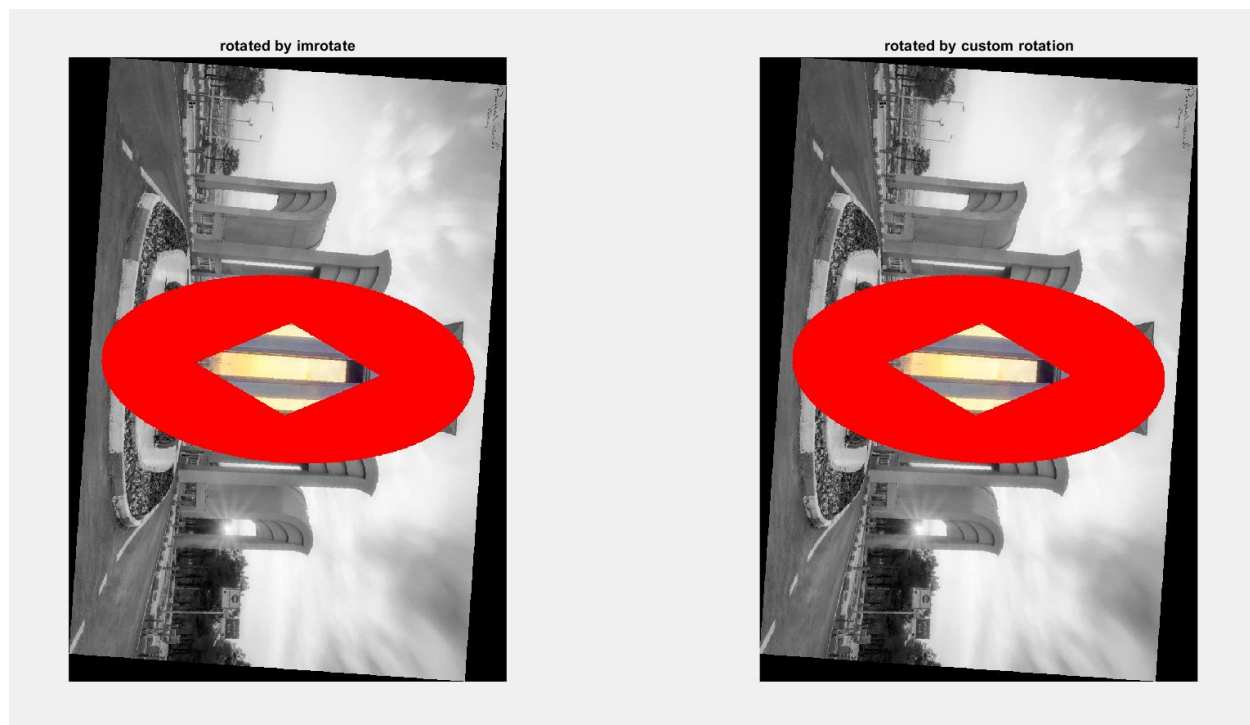
output = uint8(NewRotatedImage);
end

```

توضیح:

1. ابتدا با چرخش گوشه های مستطیل یک جعبه مرزی می سازیم. این جعبه (مستطیل جدید) مطمئنا تمامی تصویر چرخانده شده را شامل خواهد شد.
2. سپس با حرکت بر روی این تصویر جدید ایجاد شده و چرخش آن با منفی زاویه ی ورودی می توانیم بفهمیم این پیکسل در تصویر قبلی کجا قرار دارد. (اگر به صورت برعکس برویم باعث می شود نقاط سیاه در تصویر ما مشاهده شود)

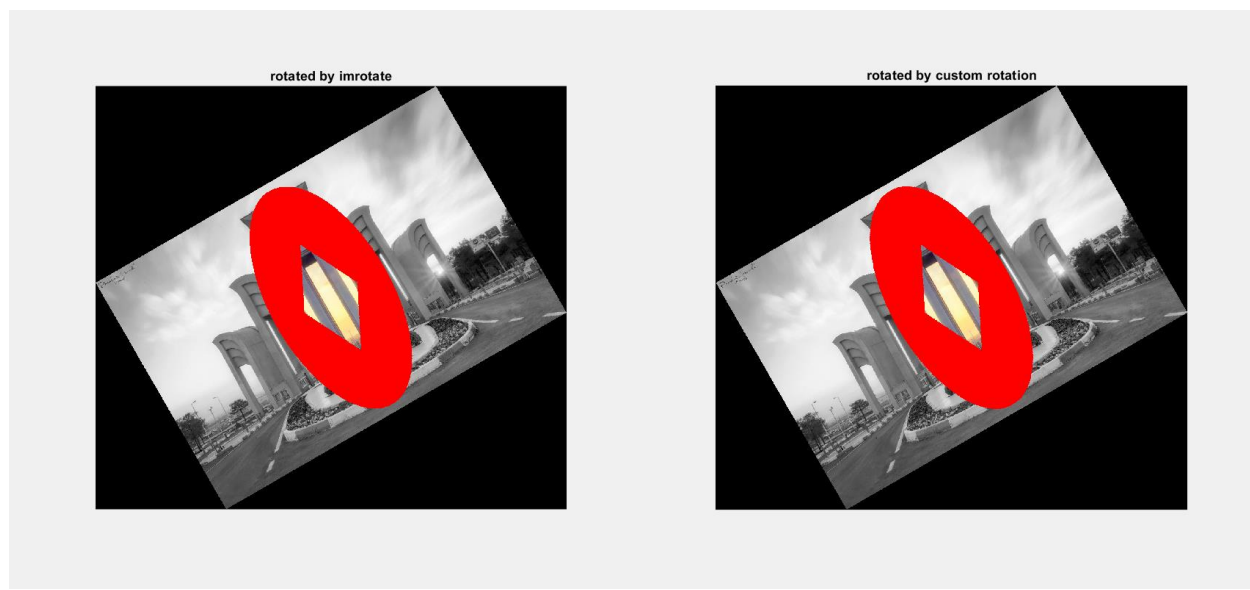
۲۶۶ درجه



۱۸۰ درجه



30 درجه



90 درجه

