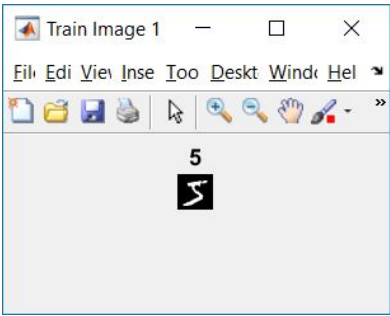
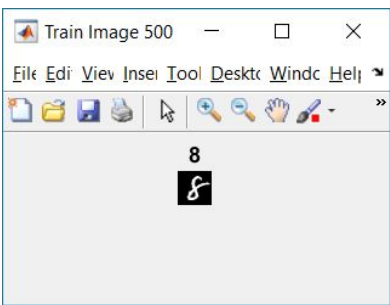
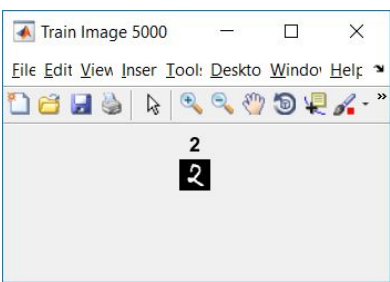


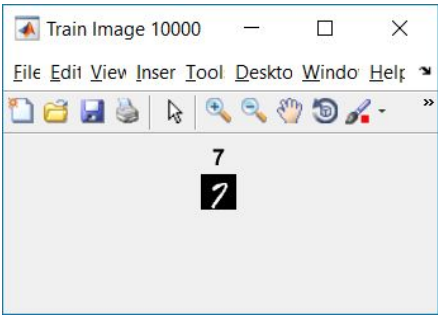
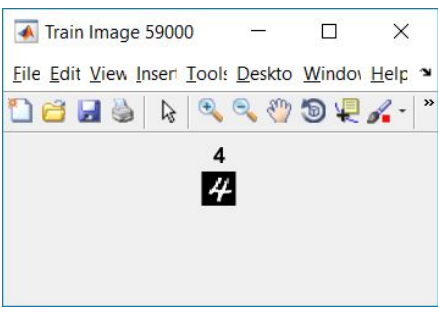
Trả lời câu hỏi

Q1: Code hiển thị ảnh với label tương ứng:

```
function ShowImgWithLabel(n, imgAll, lblAll, type)
    fprintf('Processing image %d...\n', n);
    fTitle = [type, ' Image ', num2str(n)];
    figure ('Name', fTitle, 'NumberTitle','off');
    img    = imgAll(:, n);
    img2D = reshape(img, 28, 28); %reshape
    strLabelImage = num2str(lblAll(n));
    imshow(img2D); % show image
    title(strLabelImage);
end
```

Bảng kết quả chạy thử:

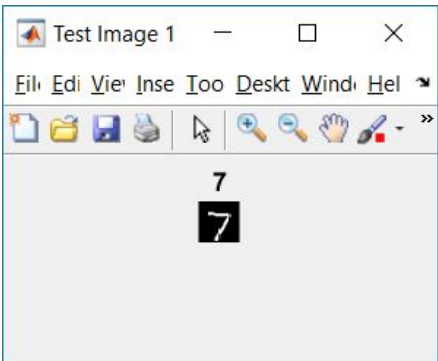
Image number	Label	Image
1	5	
500	8	
5000	2	

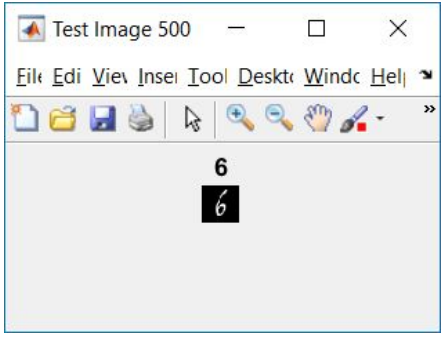
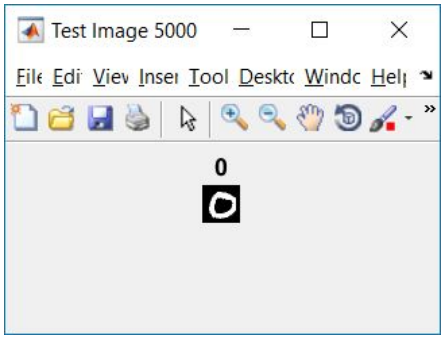
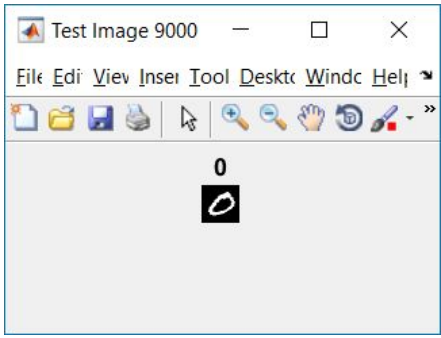
10000	7	
59000	4	

Q2: Code hiển thị ảnh với label tương ứng:

```
function ShowImgWithLabel(n, imgAll, lblAll, type)
    fprintf('Processing image %d...\n', n);
    fTitle = [type, ' Image ', num2str(n)];
    figure ('Name', fTitle, 'NumberTitle','off');
    img    = imgAll(:, n);
    img2D = reshape(img, 28, 28); %reshape
    strLabelImage = num2str(lblAll(n));
    imshow(img2D); % show image
    title(strLabelImage);
end
```

Bảng kết quả chạy thử

Image number	Label	Image
1	7	

500	6	
5000	0	
9000	0	

Q3, 4: Thống kê số lượng ảnh tương ứng với các label của tập Train và Test.

```
function ListImages = Recognition007_Digits_ListImg(ImgType)
    fprintf ('\nLoading train data...');
    lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');

    fprintf ('\nLoading test data...');
    lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');

    fprintf ('\nAll data loaded.\n');

    nTrainImages = size(lblTrainAll, 1);
    nTestImages = size(lblTestAll, 1);

    if (ImgType == "train")
        ListImages = zeros (nTrainImages, 2);
        for i = 1:nTrainImages
            ListImages(i, 1) = i;
            ListImages(i, 2) = lblTrainAll(i);
        end
    end
    csvwrite("TrainList.csv", ListImages)
```

```

elseif (ImgType == "test")
ListImages = zeros (nTestImages,2 );
for i = 1:nTestImages
    ListImages(i, 1) = i;
    ListImages(i, 2) = lblTestAll(i);
end
csvwrite("TestList.csv", ListImages)
end
end

```

Q5: Hàm trả về kết quả nhận dạng ảnh trong tập Test có thứ tự là n.

```

function TestResult = ShowTestResult(n, imgAll, lblAll, type, Mdl)
    fprintf('\nProcessing image %d...\n', n);
    img    = imgAll(:, n);
    lblImageTest    = lblAll(n);
    TestResult      = predict(Mdl, img');
End

```

n	5	500	900
result	4	6	8

Q6: Hiển thị ảnh tương ứng trong tập Test cùng kết quả nhận dạng

```

function ShowRecognizedTestImg(n, imgAll, lblAll, type, Mdl)
    fprintf('\nProcessing image %d...\n', n);
    fTitle = [type, ' Image ', num2str(n)];
    figure ('Name', fTitle, 'NumberTitle','off');
    img    = imgAll(:, n);
    lblImageTest    = lblAll(n);
    lblPredictTest = predict(Mdl, img');
    img2D = reshape(img, 28, 28); %reshape
    imshow(img2D); % show image
    strLabelImage = 'Original ';
    strLabelImage = [strLabelImage, num2str(lblImageTest), ' | '];
    strLabelImage = [strLabelImage, num2str(lblPredictTest)]
    strLabelImage = [strLabelImage, ' Predict']

    if(lblPredictTest == lblImageTest)
        strResult = 'Correct';
    else
        strResult = 'Wrong';
    end

    title(strResult);

```

```

        xlabel(strLabelImage);
end

```

Q7: Hàm đếm số lượng nhận dạng ảnh sai

```

function CountResult = WrongRecognition(n)
    CountResult = 0;

    fprintf ('\nLoading train data...');
    [imgTrainAll, lblTrainAll] = loadData('train-images.idx3-ubyte',
'train-labels.idx1-ubyte');

    fprintf ('\nLoading test data...\n');
    [imgTestAll, lblTestAll] = loadData ('t10k-images.idx3-ubyte',
't10k-labels.idx1-ubyte');

    nTestImages = size(imgTestAll, 2);

    Mdl = fitcknn(imgTrainAll', lblTrainAll);

    for i = 1:nTestImages
        img = imgTestAll(:, i);
        lblImageTest = lblTestAll(i);
        lblPredictTest = predict(Mdl, img');

        if (lblImageTest == n)
            if (lblImageTest ~= lblPredictTest)
                CountResult = CountResult + 1;
            end
        end
    end

    fprintf ('\nTotal number %d wrong recognition: %d\n', n,
CountResult);
end

```

Bảng thống kê số trường hợp chạy sai:

Number	Wrong recognition
0	7
1	6
2	40
3	40

4	38
5	32
6	14
7	36
8	54
9	42

Q7*: Hàm đếm số lượng nhận dạng ảnh sai dùng bảng confusion matrix

```
function ResultMatrix = WrongRecognitionConfusionMatrix()
    ResultMatrix = zeros(10, 2);

    fprintf ('\nLoading train data...');
    [imgTrainAll, lblTrainAll] = loadData('train-images.idx3-ubyte',
    'train-labels.idx1-ubyte');

    fprintf ('\nLoading test data...\n');
    [imgTestAll, lblTestAll] = loadData ('t10k-images.idx3-ubyte',
    't10k-labels.idx1-ubyte');

    %nTestImages = size(imgTestAll, 2);

    Mdl = fitcknn(imgTrainAll', lblTrainAll);

    fprintf ('\nRecognizing test images...\n');
    lblPredictTest = predict(Mdl, imgTestAll');

    ResultMatrix = confusionmat(lblTestAll, lblPredictTest);
end
```

Q8**: Hàm đếm số lượng nhận dạng ảnh sai dùng bảng confusion matrix và so sánh các trường hợp khác nhau của thuật toán knn.

```
function ResultMatrix = knnRecognitionConfusionMatrix(knnNumNeighbors,
knnDistance)
    ResultMatrix = zeros(10, 2);

    fprintf ('\nLoading train data...');
    [imgTrainAll, lblTrainAll] = loadData('train-images.idx3-ubyte',
    'train-labels.idx1-ubyte');

    fprintf ('\nLoading test data...\n');
```

```

[imgTestAll, lblTestAll] = loadData ('t10k-images.idx3-ubyte',
't10k-labels.idx1-ubyte');

nTestImages = size(imgTestAll, 2);

Mdl = fitcknn(imgTrainAll', lblTrainAll, 'Distance', knnDistance,
'NumNeighbors', knnNumNeighbors);

fprintf ('\nRecognizing test images...\n');
lblPredictTest = predict(Mdl, imgTestAll');

ResultMatrix = confusionmat(lblTestAll, lblPredictTest);

for i = [1:10]
ResultCorrect = ResultCorrect + ResultMatrix(i, i);
end

ResultAccurate = 100* ResultCorrect / nTestImages;

fprintf ('\nDistance Metric %s with %d-nearest neighbors
classifier', knnDistance, knnNumNeighbors);
fprintf ('\nCorrected recognition: %d of %d\n', ResultCorrect,
nTestImages);
fprintf ('Accurate rate: %.2f%%\n', ResultAccurate);
end

```

Bảng so sánh kết quả

Distance Metric	No. nearest neighbors	Accurate rate
cityblock	1	96.31%
cityblock	3	96.31%
euclidean	1	96.91%
euclidean	3	97.06%
cosine	1	97.23%
cosine	3	97.37%
chebychev	1	82.59%
chebychev	3	81.18%

spearman	1	96.85%
spearman	3	96.91%
minkowski	1	96.91%
minkowski	3	97.06%