HKUSTx: ELEC1200.2x A System View of Communications: From Signals to...

LAB 1 - TASK 2 (3/3 points)

In this task, you will implemente of the function runlength_encode that you used in Task 1. .

```
% Load the input image
lorem_img = imread('lorem_img.png');

% display the raw image
figure(1);
imshow(lorem_img);
title('Original image');


% % % % run-length encode % % %
% reshape the image to a 1D vector
image= lorem_img'; % transpose the image to vectorize row by row
img_1D = image(:)';

runs = [];
run_len = 0;
```

Correct

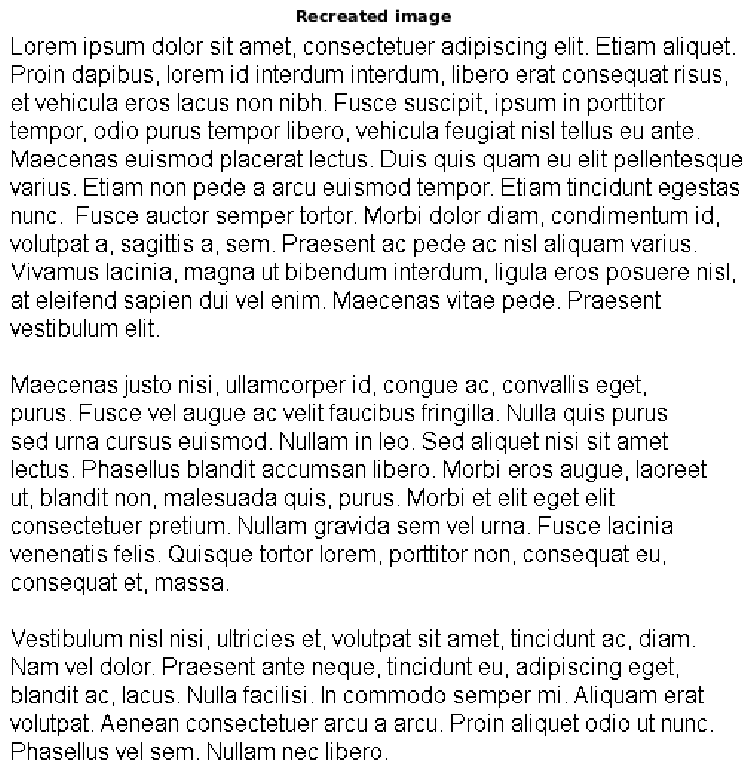
```
    % % % % Revise the following code to split runs longer than 25
5 bits  % % % %

    if run_len > 255,
        runs = [runs 255 0];
        run_len = 1;
    end
end
runs = [runs run_len];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % % % Do not change the code below % % % %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 1

tutorials

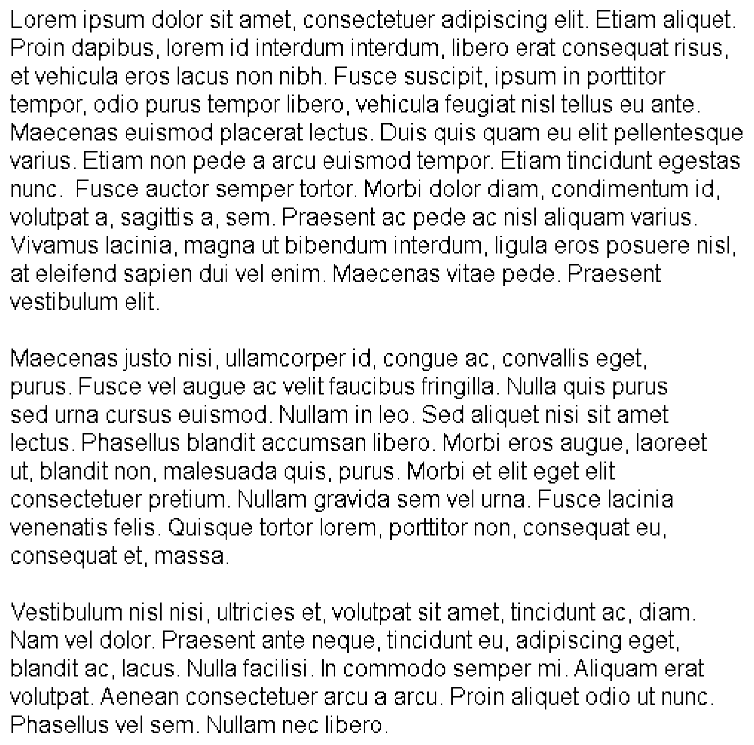

Recreated image

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam aliquet. Proin dapibus, lorem id interdum interdum, libero erat consequat risus, et vehicula eros lacus non nibh. Fusce suscipit, ipsum in porttitor tempor, odio purus tempor libero, vehicula feugiat nisl tellus eu ante. Maecenas euismod placerat lectus. Duis quis quam eu elit pellentesque varius. Etiam non pede a arcu euismod tempor. Etiam tincidunt egestas nunc. Fusce auctor semper tortor. Morbi dolor diam, condimentum id, volutpat a, sagittis a, sem. Praesent ac pede ac nisl aliquam varius. Vivamus lacinia, magna ut bibendum interdum, ligula eros posuere nisl, at eleifend sapien dui vel enim. Maecenas vitae pede. Praesent vestibulum elit.

Maecenas justo nisi, ullamcorper id, congue ac, convallis eget, purus. Fusce vel augue ac velit faucibus fringilla. Nulla quis purus sed urna cursus euismod. Nullam in leo. Sed aliquet nisi sit amet lectus. Phasellus blandit accumsan libero. Morbi eros augue, laoreet ut, blandit non, malesuada quis, purus. Morbi et elit eget elit consectetuer pretium. Nullam gravida sem vel urna. Fusce lacinia venenatis felis. Quisque tortor lorem, porttitor non, consequat eu, consequat et, massa.

Vestibulum nisl nisi, ultricies et, volutpat sit amet, tincidunt ac, diam. Nam vel dolor. Praesent ante neque, tincidunt eu, adipiscing eget, blandit ac, lacus. Nulla facilisi. In commodo semper mi. Aliquam erat volutpat. Aenean consectetuer arcu a arcu. Proin aliquet odio ut nunc. Phasellus vel sem. Nullam nec libero.

Figure 2



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam aliquet. Proin dapibus, lorem id interdum interdum, libero erat consequat risus, et vehicula eros lacus non nibh. Fusce suscipit, ipsum in porttitor tempor, odio purus tempor libero, vehicula feugiat nisl tellus eu ante. Maecenas euismod placerat lectus. Duis quis quam eu elit pellentesque varius. Etiam non pede a arcu euismod tempor. Etiam tincidunt egestas nunc. Fusce auctor semper tortor. Morbi dolor diam, condimentum id, volutpat a, sagittis a, sem. Praesent ac pede ac nisl aliquam varius. Vivamus lacinia, magna ut bibendum interdum, ligula eros posuere nisl, at eleifend sapien dui vel enim. Maecenas vitae pede. Praesent vestibulum elit.

Maecenas justo nisi, ullamcorper id, congue ac, convallis eget, purus. Fusce vel augue ac velit faucibus fringilla. Nulla quis purus sed urna cursus euismod. Nullam in leo. Sed aliquet nisi sit amet lectus. Phasellus blandit accumsan libero. Morbi eros augue, laoreet ut, blandit non, malesuada quis, purus. Morbi et elit eget elit consectetuer pretium. Nullam gravida sem vel urna. Fusce lacinia venenatis felis. Quisque tortor lorem, porttitor non, consequat eu, consequat et, massa.

Vestibulum nisl nisi, ultricies et, volutpat sit amet, tincidunt ac, diam. Nam vel dolor. Praesent ante neque, tincidunt eu, adipiscing eget, blandit ac, lacus. Nulla facilisi. In commodo semper mi. Aliquam erat volutpat. Aenean consectetuer arcu a arcu. Proin aliquet odio ut nunc. Phasellus vel sem. Nullam nec libero.

size_raw_data =

```
    250000

size_run_length =

    301688
```

You have used 1 of 10 submissions

INSTRUCTIONS

The initial MATLAB code in the above window contains a faulty implementation of the function **runlength_encode** that you used in Task 1. In this task, you will need to correct this function.

If you click the Run Code button, the initial code will generate two figures. Figure 1 displays the original image. Figure 2 shows the recreated image from the run-length code. You will observe that the recreated image does not match the original image. This is because there is an error in the implementation. If you correct it, then the two images will match.

In the following, we describe the function of the code in more detail to help you find and correct the error.

The first part of the code reads and displays the image, as in Task 1.

The next part, starting from the comment run-length encode, contains the code implementing **runlength_encode**. It starts by converting the 2D image into a 1D array, containing the binary pixel values listed out row by row.

The **for** loop then scans through this 1D array, counting the number of consecutive black or white pixels, assuming that the first run contains white pixels. It does this by counting the number of white pixels until it encounters a black pixel. It then appends the run length in the vector **runs**, and then starts counting black pixels until it reaches a white pixel, and so on. The vector **runs** contains the run lengths as decimal integers. Note that there is no loss of generality in assuming the first run contains white pixels, since if the first pixel is black, the code will just return a length of 0 for the first run of white pixels.

Once the **for** loop has finished scanning through the 1D array of pixel values, it passes the decimal vector **runs** to the function **dec2binArray**, which represents each decimal run length as an 8 bit binary number. The **reshape** command simply lists the binary numbers out as a single bit stream.

The problem in the code stems from the fact that the largest integer we can represent with 8 bits is 255, but the length of the runs of white or black pixel may exceed 255. Thus, some elements of the array **runs**, which is passed to **dec2binArray**, may exceed 255. When **dec2binArray** encounters an element greater than 255, it generates a warning message and simply encodes the number as 255 ([1 1 1 1 1 1 1 1]). Thus, when reconstructing the black and white image from the bit stream contained in the vector **run_length_code**, the function **runlength_decode** thinks the run is shorter than it actually is. This is why there are so many black pixels at the bottom of the image.

We can handle by splitting runs with length longer than 255 to separate runs of at most 255 pixels, separated by runs of opposite color pixels with length 0. For example, a run of 520 pixels would be encoded by run lengths

[255 0 255 0 10]

One way to do this is to modify the code below the line

% % % % Revise the following code to split runs longer than 255 bits% % % %

However, there are other ways. For example, instead of modifying the code within the **for** loop, you might write another for loop that scans through the final set of run lengths created by the unmodified **for** loop looking for runs longer than 255 and then handling them in some way. However, if you decide to do it, makes sure that your code generates the correct value for the decimal vector **runs** that is passed to the **dec2binArray** function.