Image and Video Processing Lab

Name: - Yash Rajput
Registration Number: 211060042 **Date: -** 09.02.2024

Experiment 4
MPEG Compression

Aim: - The objective of this experiment is to evaluate the impact of MPEG compression on image file size and visual quality. The program should prompt the user to input parameters such as the binary

Software Used: - MATLAB

Code:

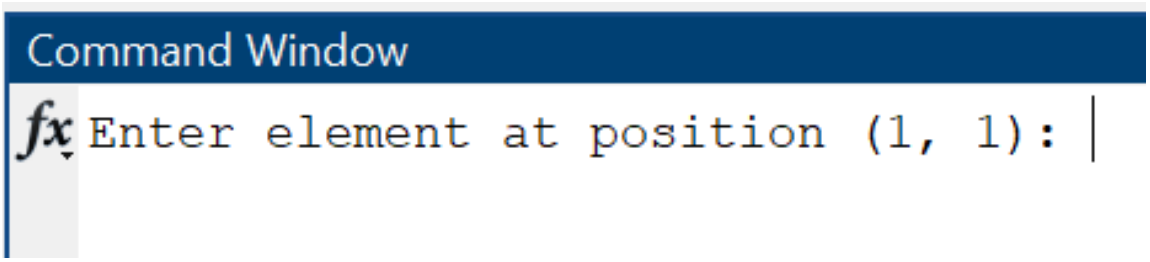
```
% IVP Lab – Experiment No: 4 – MPEG Compression
% Yash Rajput – TY EC – 211060042

clc;

% Quantization Matrix
quantization_matrix = [
    16, 11, 10, 16, 24;
    12, 12, 14, 19, 26;
    14, 13, 16, 24, 40;
    14, 17, 22, 29, 51;
    18, 22, 37, 56, 68];

% Input Matrix (example)
matrix = [
    255, 255, 0, 255, 255;
    255, 255, 0, 255, 255;
    0, 0, 0, 0, 0;
    255, 255, 0, 255, 255;
    255, 255, 0, 255, 255
];

for i = 1:5
   for j = 1:5
         Prompt user for input
        prompt = sprintf('Enter element at position (d, d): ', i, j);
        matrix(i, j) = input(prompt);
    end
end

% Perform Discrete Cosine Transform (DCT) on the input matrix
dct_matrix = dct2(matrix);

% Quantization of DCT coefficients
dct_quantized_matrix = dct_matrix ./ quantization_matrix;
dct_quantized_matrix = round(dct_quantized_matrix);

% Encode and decode the quantized matrix using run-length encoding
```

```
[encoded, decoded] = runLengthEncodeDecode(dct_quantized_matrix);

% Dequantize the decoded matrix
dequantized_block = decoded .* quantization_matrix;

% Reconstruct the image using Inverse Discrete Cosine Transform (IDCT)
output = idct2(dequantized_block);

% Display results
disp('Input Matrix:')
disp(matrix)
disp('DCT of Input Matrix:')
disp(dct_matrix)
disp('Quantization of the DCT Matrix:')
disp(dct_quantized_matrix)
disp('Encoded:')
disp(encoded);
disp('Decoded:')
disp(decoded);
disp('Dequantization of the Decoded Matrix:')
disp(dequantized_block)
disp('Final Output:')
disp(output)

% Normalize input and output images for display
normalized_matrix = matrix / max(matrix(:));
normalized_output = output / max(output(:));

% Display the input and decoded images side by side
figure;
subplot(1, 2, 1); % First subplot: Input Image
imshow(normalized_matrix);
title('Input Image');

subplot(1, 2, 2); % Second subplot: Decoded Image
imshow(normalized_output);
title('Decoded Image');

% Run-Length Encoding and Decoding Function
function [encoded, decoded] = runLengthEncodeDecode(matrix)
    % Run-Length Encoding
    flattened = matrix(:)';
    n = length(flattened);
    encoded = [];
    count = 1;

    for i = 1:n-1
        if flattened(i) == flattened(i+1)
            count = count + 1;
        else
            encoded = [encoded flattened(i) count];
            count = 1;
        end
    end
    encoded = [encoded flattened(end) count];

    % Run-Length Decoding
    decoded = [];
    i = 1;

    while i <= length(encoded)
        value = encoded(i);
        count = encoded(i+1);
        decoded = [decoded repmat(value, 1, count)];
```

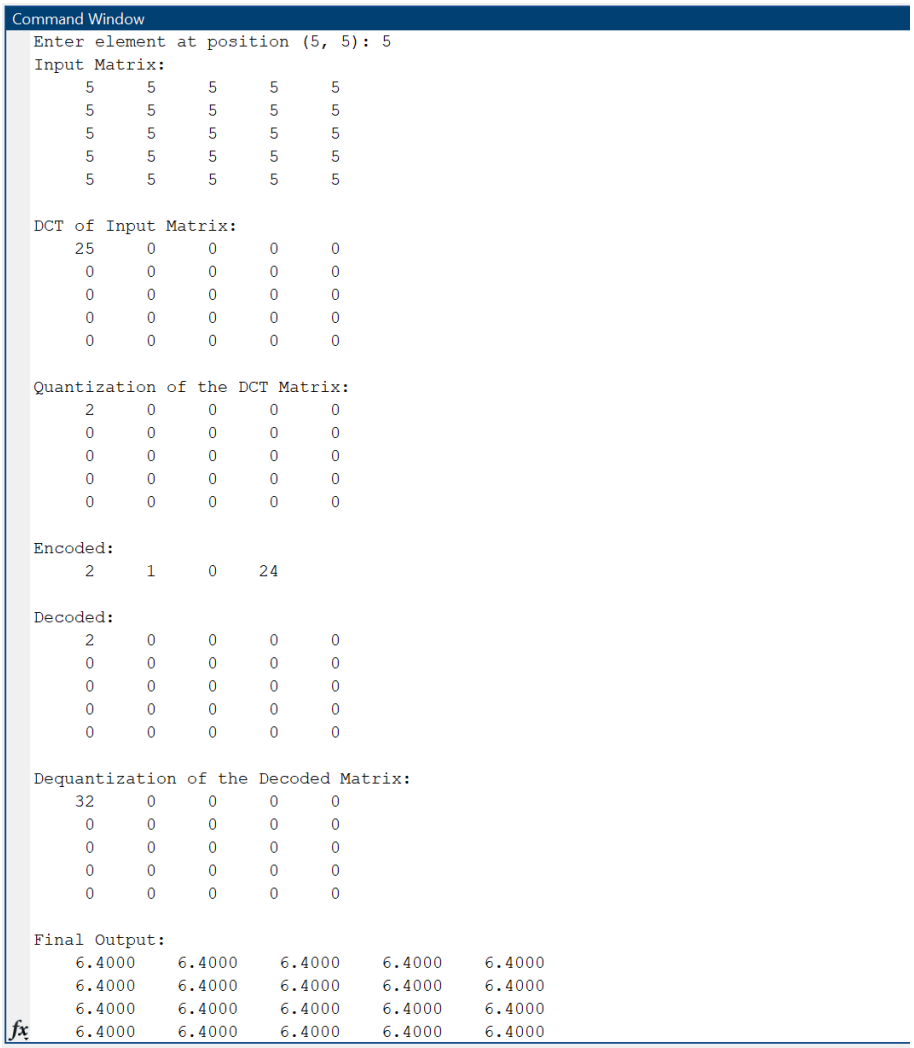
```
        i = i + 2;
    end

    % Reshape decoded array back into a matrix
    decoded = reshape(decoded, size(matrix));
end
```

User Interface:

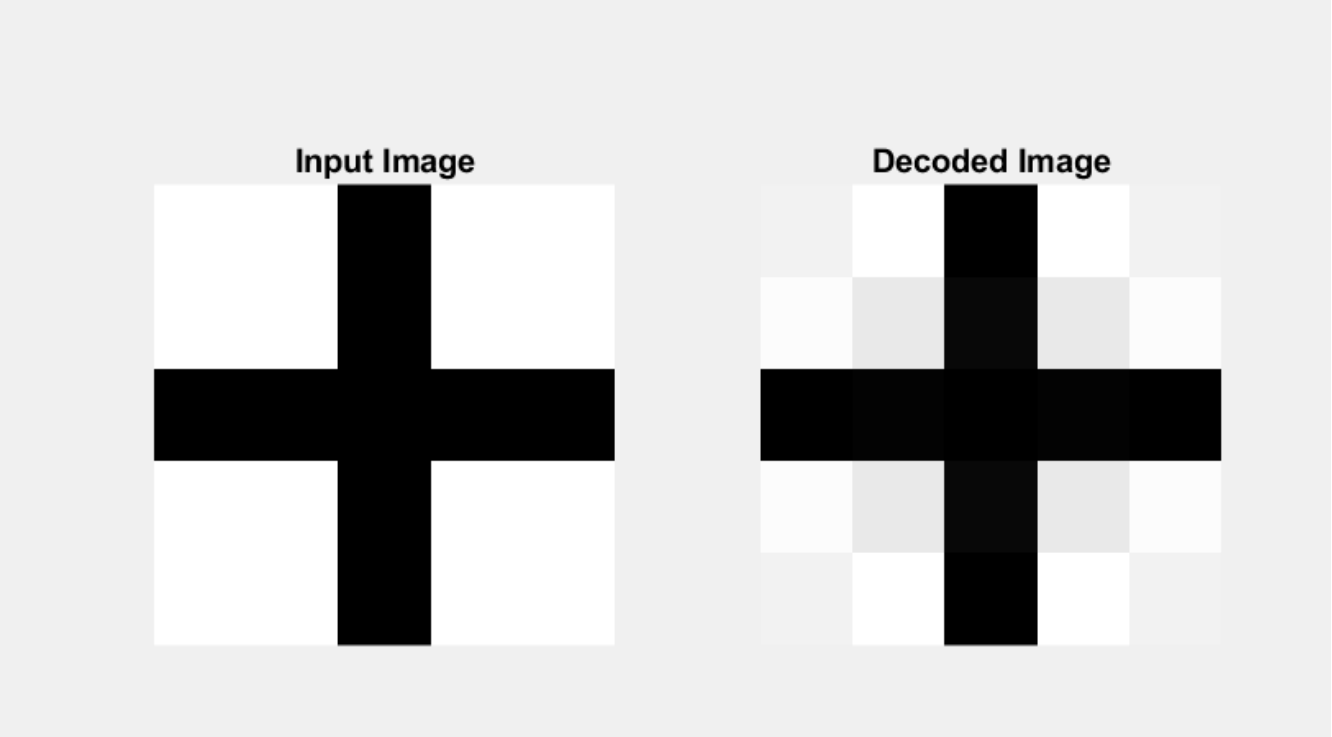
Step 1: The code begins with asking the user for the input matrix



Step 2: After the input of the matrix your command window will show all the necessary blocks and matrices

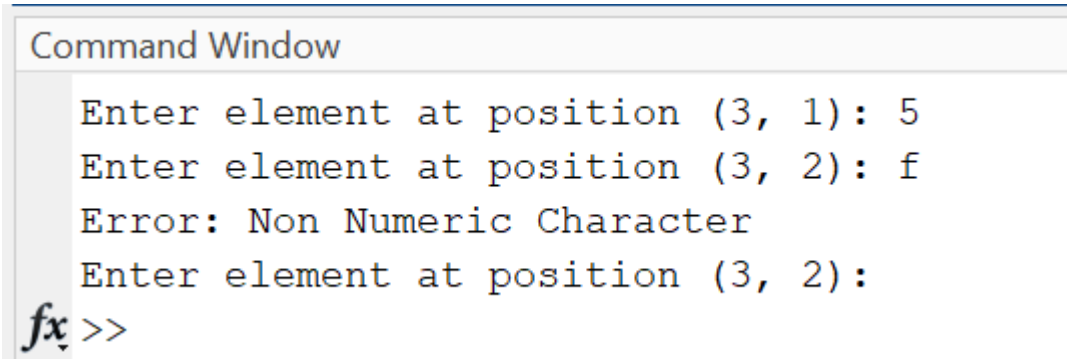


Step 3: A figure will pop showing you the Input image and image after the Run Length encoding-decoding

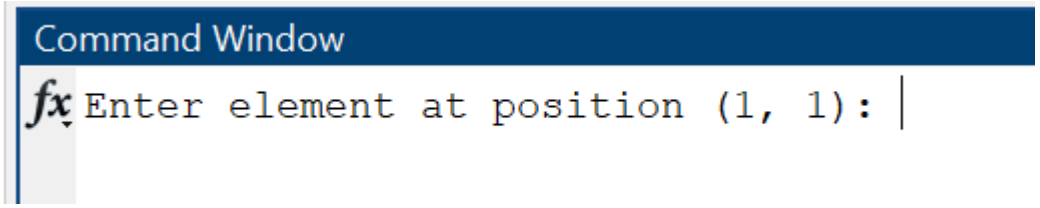


Error Handling:

Case 1: The Entered Matrix contains Non-Numeric Character



Case 2: Asymmetric Matrix or Matrix of not 5x5 shape



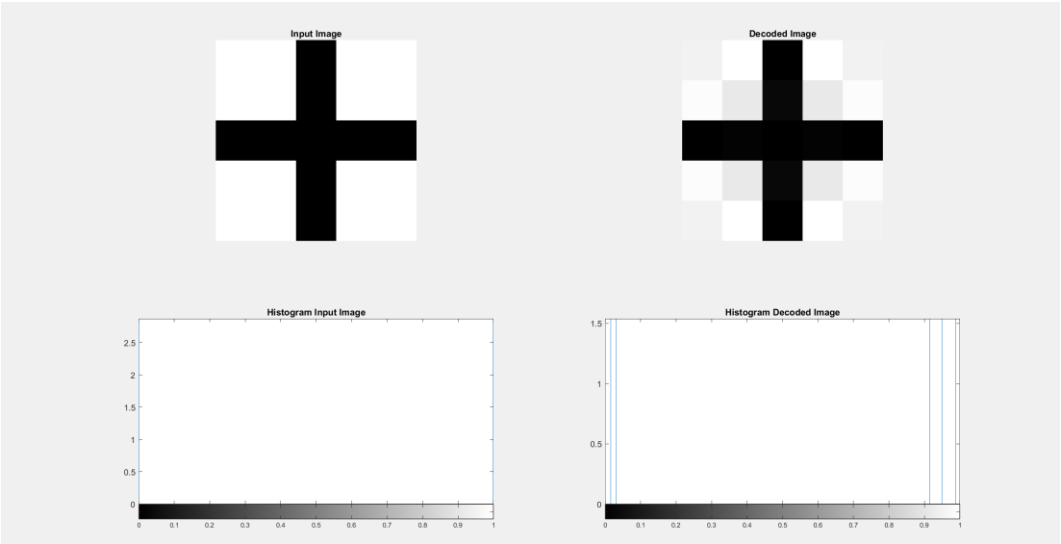
This following case is not possible as the code requires user to enter the numbers one by one till the matrix does not complete

Output Analysis

Binary Test Case 1

```
Input Matrix:
   255   255     0   255   255
   255   255     0   255   255
     0     0     0     0     0
   255   255     0   255   255
   255   255     0   255   255

DCT of Input Matrix:
  816.0000         0  288.4996         0 -288.4996
         0         0         0         0         0
  288.4996         0  102.0000         0 -102.0000
         0         0         0         0         0
 -288.4996         0 -102.0000         0  102.0000

Quantization of the DCT Matrix:
    51     0    29     0   -12
     0     0     0     0     0
    21     0     6     0    -3
     0     0     0     0     0
   -16     0    -3     0     1

Encoded:
  Columns 1 through 21

    51     1     0     1    21     1     0     1   -16     1     0     5    29     1     0     1     6     1     0     1    -3

  Columns 22 through 34

     1     0     5   -12     1     0     1    -3     1     0     1     1     1

Decoded:
    51     0    29     0   -12
     0     0     0     0     0
    21     0     6     0    -3
     0     0     0     0     0
   -16     0    -3     0     1

Dequantization of the Decoded Matrix:
   816     0   290     0  -288
     0     0     0     0     0
   294     0    96     0  -120
     0     0     0     0     0
  -288     0  -111     0    68


Final Output:
  251.1198  265.1131   -5.9542  265.1131  251.1198
  261.8357  242.3290    8.6953  242.3290  261.8357
   -3.9760    3.8886   -6.8975    3.8886   -3.9760
  261.8357  242.3290    8.6953  242.3290  261.8357
  251.1198  265.1131   -5.9542  265.1131  251.1198
```

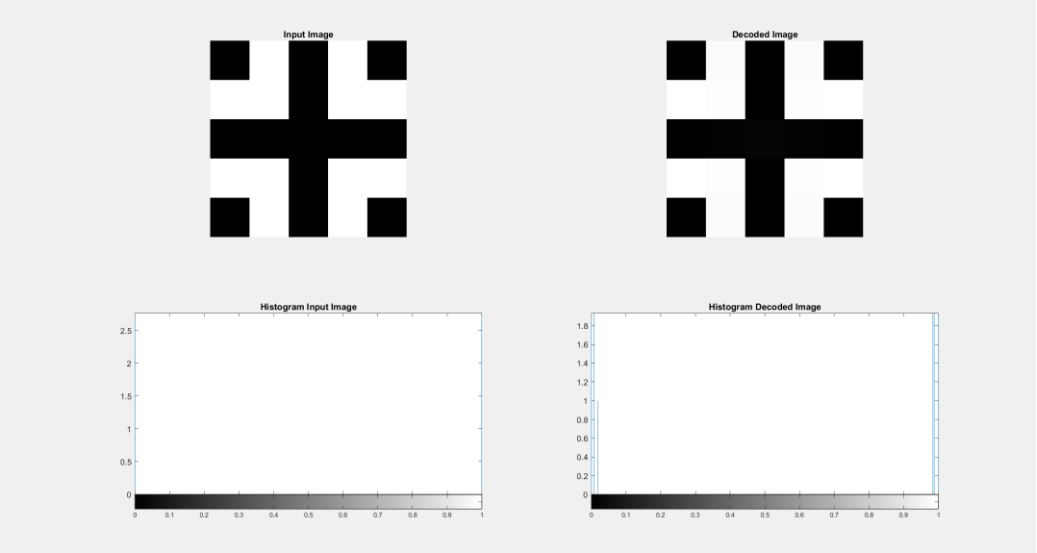
Visual Representation:



Binary Test Case 2

```
Input Matrix:
     0   255     0   255     0
   255   255     0   255   255
     0     0     0     0     0
   255   255     0   255   255
     0   255     0   255     0

DCT of Input Matrix:
  612.0000         0   55.0985         0 -377.6508
         0         0         0         0         0
   55.0985         0 -165.0395         0 -204.0000
         0         0         0         0         0
 -377.6508         0 -204.0000         0   63.0395

Quantization of the DCT Matrix:
    38     0     6     0   -16
     0     0     0     0     0
     4     0   -10     0    -5
     0     0     0     0     0
   -21     0    -6     0     1

Encoded:
  Columns 1 through 21

    38     1     0     1     4     1     0     1   -21     1     0     5     6     1     0     1   -10     1     0     1    -6

  Columns 22 through 34

     1     0     5   -16     1     0     1    -5     1     0     1     1     1

Decoded:
    38     0     6     0   -16
     0     0     0     0     0
     4     0   -10     0    -5
     0     0     0     0     0
   -21     0    -6     0     1

Dequantization of the Decoded Matrix:
   608     0    60     0  -384
     0     0     0     0     0
    56     0  -160     0  -200
     0     0     0     0     0
  -378     0  -222     0    68

Final Output:
    0.0512  254.0405   -1.3047  254.0405    0.0512
  258.3274  255.3167  -11.2828  255.3167  258.3274
   -7.9241    2.4077    5.2641    2.4077   -7.9241
  258.3274  255.3167  -11.2828  255.3167  258.3274
    0.0512  254.0405   -1.3047  254.0405    0.0512
```

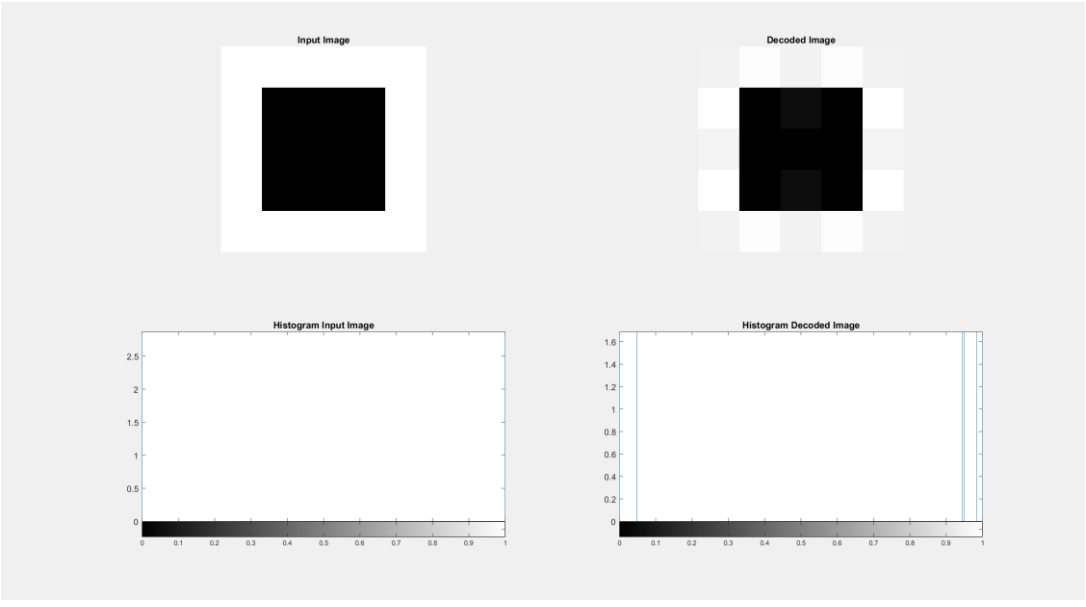
Visual Representation:



Binary Test Case 3

```
Input Matrix:
   255   255   255   255   255
   255     0     0     0   255
   255     0     0     0   255
   255     0     0     0   255
   255   255   255   255   255

DCT of Input Matrix:
  816.0000         0  350.1016         0  133.7269
         0         0         0         0         0
  350.1016         0 -267.0395         0 -102.0000
         0         0         0         0         0
  133.7269         0 -102.0000         0  -38.9605

Quantization of the DCT Matrix:
    51     0    35     0     6
     0     0     0     0     0
    25     0   -17     0    -3
     0     0     0     0     0
     7     0    -3     0    -1

Encoded:
  Columns 1 through 21

    51     1     0     1    25     1     0     1     7     1     0     5    35     1     0     1   -17     1     0     1    -3

  Columns 22 through 34

     1     0     5     6     1     0     1    -3     1     0     1    -1     1

Decoded:
    51     0    35     0     6
     0     0     0     0     0
    25     0   -17     0    -3
     0     0     0     0     0
     7     0    -3     0    -1

Dequantization of the Decoded Matrix:
   816     0   350     0   144
     0     0     0     0     0
   350     0  -272     0  -120
     0     0     0     0     0
   126     0  -111     0   -68

Final Output:
  250.0682  260.4158  250.5391  260.4158  250.0682
  264.0954  -11.0570   12.8081  -11.0570  264.0954
  251.0461   -0.4269   -2.0224   -0.4269  251.0461
  264.0954  -11.0570   12.8081  -11.0570  264.0954
  250.0682  260.4158  250.5391  260.4158  250.0682
```

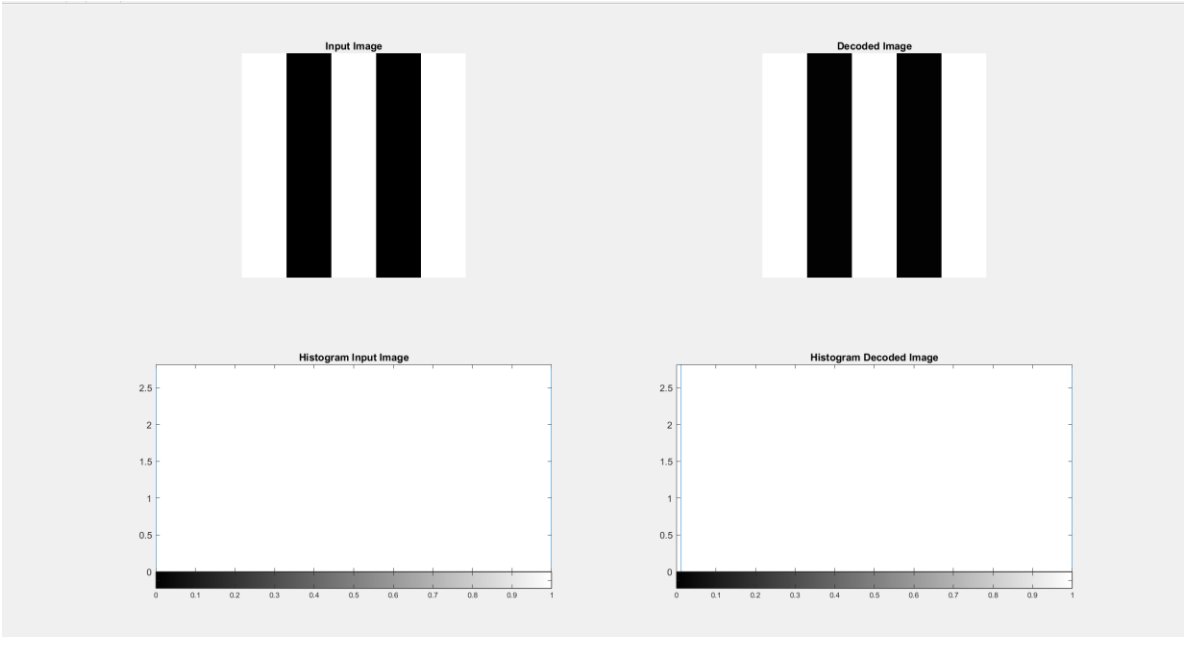
Visual Representation:



Binary Test Case 4

```
Command Window
  Input Matrix:
     255     0   255     0   255
     255     0   255     0   255
     255     0   255     0   255
     255     0   255     0   255
     255     0   255     0   255

  DCT of Input Matrix:
    765.0000         0  222.8782         0  583.5026
           0         0         0         0         0
           0         0         0         0         0
           0         0         0         0         0
           0         0         0         0         0

  Quantization of the DCT Matrix:
      48     0    22     0    24
       0     0     0     0     0
       0     0     0     0     0
       0     0     0     0     0
       0     0     0     0     0

  Encoded:
      48     1     0     9    22     1     0     9    24     1     0     4

  Decoded:
      48     0    22     0    24
       0     0     0     0     0
       0     0     0     0     0
       0     0     0     0     0
       0     0     0     0     0

  Dequantization of the Decoded Matrix:
     768     0   220     0   576
       0     0     0     0     0
       0     0     0     0     0
       0     0     0     0     0
       0     0     0     0     0

  Final Output:
    254.2856    2.5683  254.2920    2.5683  254.2856
    254.2856    2.5683  254.2920    2.5683  254.2856
    254.2856    2.5683  254.2920    2.5683  254.2856
    254.2856    2.5683  254.2920    2.5683  254.2856
    254.2856    2.5683  254.2920    2.5683  254.2856
fx
```

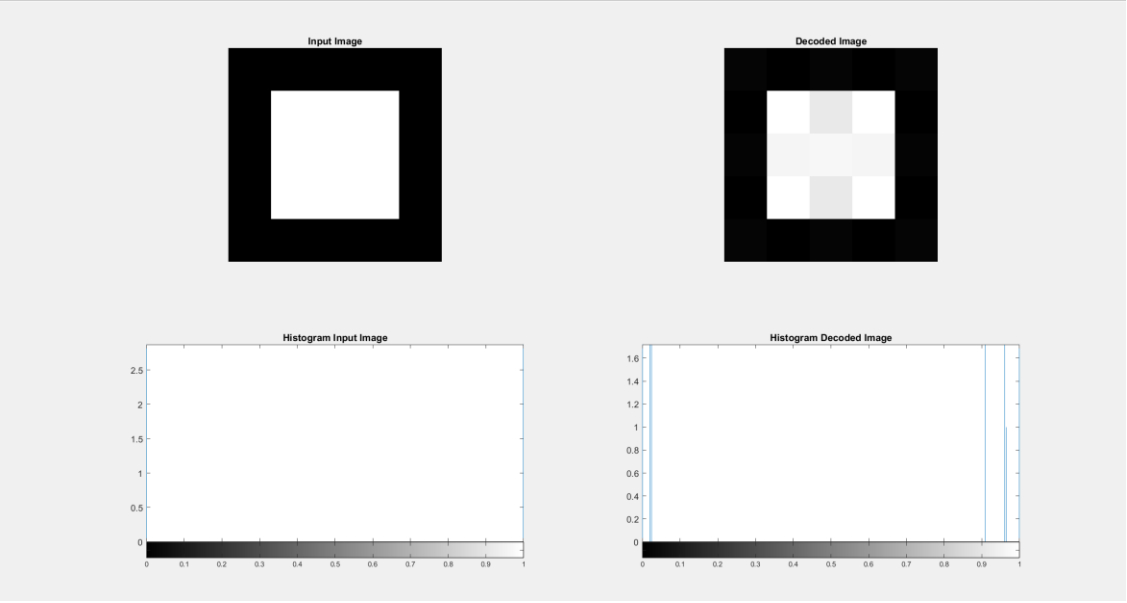
Visual Representation:



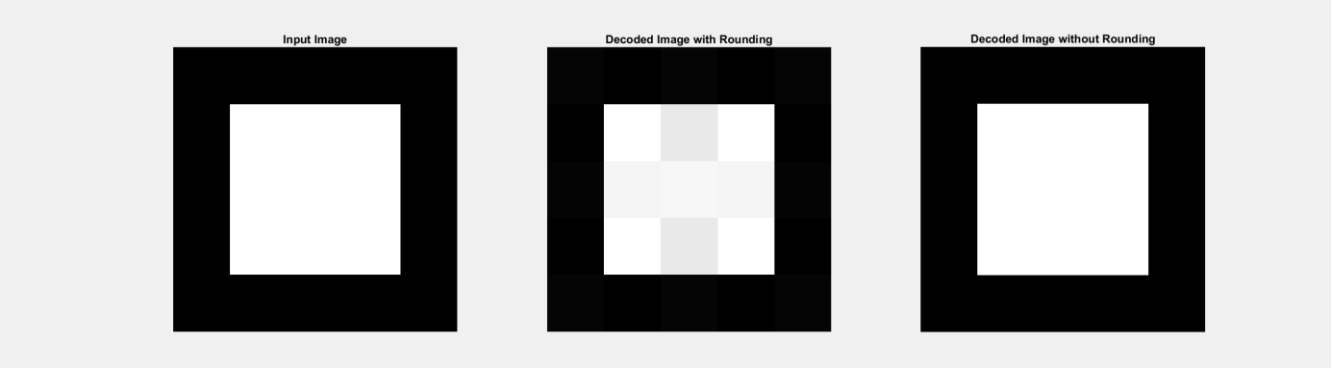
Binary Test Case 5

```
Input Matrix:
     0     0     0     0     0
     0   255   255   255     0
     0   255   255   255     0
     0   255   255   255     0
     0     0     0     0     0

DCT of Input Matrix:
  459.0000         0 -350.1016         0 -133.7269
         0         0         0         0         0
 -350.1016         0  267.0395         0  102.0000
         0         0         0         0         0
 -133.7269         0  102.0000         0   38.9605

Quantization of the DCT Matrix:
    29     0   -35     0    -6
     0     0     0     0     0
   -25     0    17     0     3
     0     0     0     0     0
    -7     0     3     0     1

Encoded:
  Columns 1 through 21

    29     1     0     1   -25     1     0     1    -7     1     0     5   -35     1     0     1    17     1     0     1     3

  Columns 22 through 34

     1     0     5    -6     1     0     1     3     1     0     1     1     1

Decoded:
    29     0   -35     0    -6
     0     0     0     0     0
   -25     0    17     0     3
     0     0     0     0     0
    -7     0     3     0     1

Dequantization of the Decoded Matrix:
   464     0  -350     0  -144
     0     0     0     0     0
  -350     0   272     0   120
     0     0     0     0     0
  -126     0   111     0    68

Final Output:
    5.9318   -4.4158    5.4609   -4.4158    5.9318
   -8.0954  267.0570  243.1919  267.0570   -8.0954
    4.9539  256.4269  258.0224  256.4269    4.9539
   -8.0954  267.0570  243.1919  267.0570   -8.0954
    5.9318   -4.4158    5.4609   -4.4158    5.9318
```

Visual Representation:



NOTE: Information is in the white space which can be seen a bit different due to the error due to round off function, if we do not round it, we can see complete match of input image and decoded image



Conclusion: This code outlines a simplified MPEG (Moving Picture Experts Group) compression pipeline. The process involves quantization of the input matrix followed by discrete cosine transform (DCT), and then applies run-length encoding for compression.

Here's a rephrased description of the process:

1. Quantization:
The input image matrix (`matrix`) is quantized using a predefined quantization matrix (`quantization_matrix`). Quantization reduces the precision of the image data, emphasizing important information while discarding less critical details.

2. Discrete Cosine Transform (DCT):
The quantized matrix undergoes DCT (`dct2(matrix)`) to convert spatial pixel information into frequency coefficients (`dct_matrix`). DCT helps in concentrating the image energy into a smaller set of coefficients, aiding in compression.

3. Run-Length Encoding (RLE):
After DCT, the resulting coefficients are processed through run-length encoding (`runLengthEncodeDecode`). RLE efficiently represents consecutive identical values by encoding the sequence of coefficients and their counts.

The overall process aims to compress the image data by transforming it into the frequency domain (DCT), reducing data precision through quantization, and further compressing the transformed coefficients using run-length encoding. This sequence mirrors fundamental aspects of MPEG compression, although a complete MPEG system would involve additional stages like motion estimation, entropy coding, and possibly inter-frame compression techniques for video sequences.