

```

%created by- Saurabh Singh
%date- 31/01/2026
clc;
clear all;
close all;

inputImage = imread("I.jpg");

if size(inputImage,3)==3
    grayImage = rgb2gray(inputImage);
else
    grayImage = inputImage;
end

figure;
imshow(grayImage);
title('Grayscale Image');

% Histogram and probability calculation
pixelCount = imhist(grayImage);
probability = pixelCount / sum(pixelCount);

% Keep only symbols that appear
symbolValues = find(probability > 0) - 1;
symbolProb = probability(probability > 0);

% Cumulative probability intervals
cumulativeProb = cumsum(symbolProb);
lowerBoundProb = [0; cumulativeProb(1:end-1)];
upperBoundProb = cumulativeProb;

% Select small sequence for demo
symbolSequence = symbolValues(1:10);

% Initialize interval
lowRange = 0;
highRange = 1;

for n = 1:length(symbolSequence)

    currentSymbol = symbolSequence(n);
    symbolIndex = find(symbolValues == currentSymbol);

    intervalWidth = highRange - lowRange;

    highRange = lowRange + intervalWidth * upperBoundProb(symbolIndex);
    lowRange = lowRange + intervalWidth * lowerBoundProb(symbolIndex);

end

encodedValue = (lowRange + highRange)/2;

disp("Arithmetic Coding Output:");
fprintf("Final Lower Bound = %.10f\n", lowRange);
fprintf("Final Upper Bound = %.10f\n", highRange);
fprintf("Encoded Tag Value = %.10f\n", encodedValue);

```

Arithmetic Coding Output:

Final Lower Bound = 0.0000038624

Final Upper Bound = 0.0000038624

Encoded Tag Value = 0.0000038624

Grayscale Image

