```
clc;
clear all;
close all;

I=imread("input.jpeg");
if size(I,3)==3
    I=rgb2gray(I);
end
figure
imshow(I);

counts=imhist(I);
p=counts/sum(counts);
%Computing probabilities of gray levels using histogram normalization.

symbols=find(p>0)-1;
p=p(p>0);
%Keeping only those symbols that actually appear in the image.
%keeping those having probability greater than 0

cumProb=cumsum(p);
low=[0; cumProb(1:end-1)];
high=cumProb;
%Generating cumulative probability intervals required for arithmetic
 coding.

sequence=symbols(1:10);
%Taking a small symbol sequence from the image for encoding
 demonstration.

L=0;
H=1;
%Initializing the arithmetic coding interval as [0,1].

for k=1:length(sequence)

    sym=sequence(k);
    idx=find(symbols==sym);
    %Finding the index of the current symbol in the probability table.

    range=H-L;
    %Computing the current interval width.

    H=L+range*high(idx);
    L=L+range*low(idx);
    %Updating interval boundaries based on symbol probability
 subrange.

end

code=(L+H)/2;
%Choosing a number inside final interval as the arithmetic code.
```

```
disp("Arithmetic Coding Output:");
fprintf("Final Lower Bound=%.10f\n",L);
fprintf("Final Upper Bound=%.10f\n",H);
fprintf("Encoded Tag Value=%.10f\n",code);
```

```
Arithmetic Coding Output:
Final Lower Bound=0.0000005185
Final Upper Bound=0.0000005185
Encoded Tag Value=0.0000005185
```

Published with MATLAB® R2021a