

```

clc;
clearvars;
close all;

p = input('Enter the probabilities: ');
n = length(p);
symbols = [1:n];
[dict, avglen] = huffmandict(symbols, p);
temp = dict;
t = dict(:,2);

for i = 1:length(temp)
    temp{i,2} = num2str(temp{i,2});
end

disp('The Huffman code dict is: ');
disp(temp)

% Encoder
fprintf('Enter a symbol between 1 to %d as array:', n);
sym = input(' ');
encod = huffmanenco(sym, dict);
disp('The encoded output: ');
disp(encod);

% Decoder
bits = input('Enter the encoded bit stream as array: ');
decod = huffmandeco(bits, dict);
fprintf('The decoded symbols are: %d\n', decod);

H = 0;
for k = 1:n
    H = H + (p(k)*log2(1/p(k)));
end
fprintf('Entropy is %f bits\n', H);
N = H / avglen;
fprintf('Efficiency is: %f\n', N);
for r = 1:n
    l(r) = length(t{r});
end
m = max(l);
s = min(l);
v = m-s; % Variance
fprintf('The variance is: %d\n', v);

```