Shannon Fano Method:-

```
clc;
clear;
close all;
printf("Name: Jay Kotwal\n");
printf("Roll No: 32137\n");
printf("Batch: L6\n");
pkg load communications
 % Load the Communications Package
symbols = {'A', 'B', 'C', 'D', 'E'};
probabilities = [0.3, 0.2, 0.3, 0.1, 0.1];
% Create Shannon Fano Dictionary
symbols_numeric = 1:length(symbols);
dict = shannonfanodict(symbols_numeric, probabilities);
% Encode a sequence using Shannon Fano
sequence = [1,2,3,4,5];
 encoded_sequence = shannonfanoenco (sequence, dict);
% Decode the sequence using Shannon Fano Coding
decoded_sequence = shannonfanodeco(encoded_sequence, dict);
% Convert numeric symbols back to original symbols
decoded_symbols = symbols(decoded_sequence);
% Display Shannon Fano Codes
disp('Shannon Fano Codes:');
for i = 1:length(symbols)
 fprintf('%s: %s\n', symbols{i}, mat2str(dict{1, i}));
end
% Calculate Entropy
entropy = @(p) -sum(p.* log2(p));
 H_X = entropy(probabilities);
% Calculate Average Code Length
 code_lengths = cellfun(@length, dict(:, 2));
 average_code_length = sum(probabilities.* code_lengths);
% Calculate Efficiency
efficiency = H_X / average_code_length;
% Display Entropy and Efficiency
fprintf('Entropy H(X): %.4f bits\n', H_X);
fprintf('Average Code Length: %.4f bits\n', average_code_length);
fprintf('Efficiency: %.4f\n', efficiency);
% Display Encoded and Decoded Sequence
disp('Encoded Sequence:');
disp(encoded_sequence);
disp('Decoded Symbols:');
disp(decoded_symbols);
```

OUTPUT

Name: Jay Kotwal
Roll No: 32137
Batch: L6
Shannon Fano Codes:
A: [0 0]
B: [1 0 0]
C: [0 1]
D: [1 1 0 0]
E: [1 1 1 0]
Entropy H(X): 2.1710 bits
Average Code Length: 3.0000 bits
Efficiency: 0.7237
Encoded Sequence:
  0 0 1 0 0 0 1 1 1 0 0 1 1 1 0
Decoded Symbols:
{
  [1,1] = A
  [1,2] = B
  [1,3] = C
  [1,4] = D
  [1,5] = E
}