```
% Shannon Fano Coding
clear all;
close all;
clc;

% Define corresponding frequencies
symbols = {'0-30', '31-59', '60-63', '64-100', '101-127', '128-150',
'151-200', '201-255'};
frequencies = [2048, 2048, 2048, 2048, 819, 819, 3277, 3277];

% Calculate probabilities
probabilities = frequencies / sum(frequencies);

% Sort probabilities in descending order
[probabilities, idx] = sort(probabilities, 'descend');
symbols = symbols(idx);

% Initialize variables
codes = cell(size(symbols));
code_lengths = zeros(size(symbols));

% Use a stack to implement the Shannon-Fano algorithm
stack = {{symbols, probabilities, ''}};


while ~isempty(stack)
    data = stack{end};
    stack(end) = [];

    s = data{1};
    p = data{2};
    pre = data{3};

    % If only one symbol remains, assign the final code
    if length(s) == 1
        idx = strcmp(symbols, s{1});
        codes{idx} = pre;
        code_lengths(idx) = length(pre);
        continue;
    end

    % Find the best split point where cumulative probability is close to half
    split = find(cumsum(p) >= sum(p) / 2, 1);

    % Push the two partitions
    stack{end+1} = {s(1:split), p(1:split), strcat(pre, '0')};
    stack{end+1} = {s(split+1:end), p(split+1:end), strcat(pre, '1')};
end

% Display
fprintf('Symbol\tCode\tLength\n');
for i = 1:length(symbols)
```

```
    fprintf('%s\t%s\t%d\n', symbols{i}, codes{i}, code_lengths(i));
end
```

```
Symbol    Code    Length
151-200    000    3
201-255    001    3
0-30    01    2
31-59    100    3
60-63    101    3
64-100    110    3
101-127    1110    4
128-150    1111    4
```

Published with MATLAB® R2024b