```
%shanon-Fano coding
clc;
clear;
close all;
% Input symbols and probabilities
symbols = {'m1','m2','m3','m4','m5','m6','m7','m8'};

% Example probabilities (CHANGE THESE)
P = [0.22 0.18 0.15 0.12 0.10 0.09 0.08 0.06];

% Normalize (safety check)
P = P / sum(P);

% Sort symbols by descending probability
[P_sorted, idx] = sort(P, 'descend');
symbols_sorted = symbols(idx);

% Initialize codes
codes = repmat({''}, 1, length(P_sorted));

% Shannon–Fano recursive function
function codes = shannon_fano(P, symbols, prefix)
    if length(P) == 1
        codes = {prefix};
        return;
    end

    % Find partition with minimum probability difference
    total = sum(P);
    diff = inf;
    split = 1;

    for i = 1:length(P)-1
        d = abs(sum(P(1:i)) - sum(P(i+1:end)));
        if d < diff
            diff = d;
            split = i;
        end
    end

    % Recursive coding
    left  = shannon_fano(P(1:split), symbols(1:split), [prefix '0']);
    right = shannon_fano(P(split+1:end), symbols(split+1:end), [prefix '1']);

    codes = [left right];
end

% Generate codes
codes_sorted = shannon_fano(P_sorted, symbols_sorted, '');

% Display results
disp('Shannon-Fano Coding');
disp('--------------------');
for i = 1:length(symbols_sorted)
    fprintf('%s  P=%.3f  Code=%s\n', ...
```

```
        symbols_sorted{i}, P_sorted(i), codes_sorted{i});
end
```

```
Shannon-Fano Coding
-------------------
m1  P=0.220  Code=00
m2  P=0.180  Code=010
m3  P=0.150  Code=011
m4  P=0.120  Code=100
m5  P=0.100  Code=101
m6  P=0.090  Code=110
m7  P=0.080  Code=1110
m8  P=0.060  Code=1111
```

Published with MATLAB® R2024a