```
clear all
close all
clc

%% This script generates the 'Hamming Index Calc Function' for viterbi decoder

n = 2;
k = 1; %only support k = 1 now!!!!!!!!!!!!
m = 7;
x_ox = [171; 133];
x = oct2dec(x_ox);
generator = dec2bin(x, m) - 48;
payload = 256*8;
tb_length = 128;
nu = m - 1;
state = 2^nu;
delete hamming_index_*.v;

for radix = [2, 4, 8, 16]
    s = sprintf('hamming_index_%d.v', radix);
    fid = fopen(s, 'a');
    fprintf(fid, 'function integer hamming_index_%d;\n', radix);
    fprintf(fid, 'input integer a;\n');
    fprintf(fid, 'input integer b;\n\n');
    fprintf(fid, 'case(a)\n');
    for cnt1 = 0 : state-1
        fprintf(fid, '\t%d : \n\t\tcase(b)\n', cnt1);
        for cnt2 = 0 : radix - 1
            branch_metric_index = mod(cnt1*radix+cnt2, state);
            hamm_dist_index = next_state_output(branch_metric_index, cnt1, generator, nu, radix, n);
            fprintf(fid, '\t\t\t%d : hamming_index = %d;\n', branch_metric_index, hamm_dist_index);
        end
    end
    fprintf(fid, '\t\tendcase\n');
    fprintf(fid, 'endcase\n');
    fprintf(fid, 'endfunction\n');
    fclose(fid);
end

%% function definitions are not permitted in scripts, seperate it when using the scripts above
% function result = next_state_output(present_state, next_state, generator, nu, radix, n)
% %function [result result_str] = next_state_output(present_state, next_state, generator, nu, radix, n)
%
% p = dec2bin(present_state, nu) - 48;
% n = dec2bin(next_state, nu) - 48;
% r = log2(radix);
% reg = [p, 0];
% result = 0;
% for cnt = r : -1 : 1
%     reg = [n(cnt), reg(1 : end-1)];
%     result_bi = char(mod(sum((generator .* kron(reg, ones(2,1))), 2)',2) + 48);
%     res_temp = bin2dec((result_bi));
%     result = result * 4 + res_temp;
% end;

```