Exp8.m

```
clc;
clear all;
close all;

% Input parameters
n = input('Enter the value of n: ');
k = input('Enter the value of k: ');
m = n - k;

% Generate generator polynomial for the cyclic code
G = cyclpoly(n, k, 'max');
disp('Generator polynomial:');
disp(poly2sym(G));

% Generate codewords using the generator polynomial
d1 = [1 0 0 0];
c1 = poly2sym(d1) * poly2sym(G);
d2 = [0 1 0 0];
c2 = poly2sym(d2) * poly2sym(G);
d3 = [0 0 1 0];
c3 = poly2sym(d3) * poly2sym(G);
d4 = [0 0 0 1];
c4 = poly2sym(d4) * poly2sym(G);

% Display codewords and data vectors
disp('Data vectors:');
d = [d1; d2; d3; d4]
disp('Codewords:');
s = [c1; c2; c3; c4]

% Generate parity-check matrix and syndrome table for Hamming code
parmat = hammgen(m);
trt = syndtable(parmat);

% First received codeword
recd = [0 1 0 1 0 0 0];
syndrome = rem(recd * parmat', 2);
syndrome_de = bi2de(syndrome, 'left-msb');
disp(['Syndrome (decimal): ', num2str(syndrome_de)]);
disp(['Syndrome (binary): ', num2str(syndrome)]);
Error = trt(1 + syndrome_de, :);
corrected_code = rem(Error + recd, 2);
disp('Corrected code for first received word:');
disp(corrected_code);

% Second received codeword
recd = [1 1 0 1 1 0 1];
syndrome = rem(recd * parmat', 2);
syndrome_de = bi2de(syndrome, 'left-msb');
disp(['Syndrome (decimal): ', num2str(syndrome_de)]);
disp(['Syndrome (binary): ', num2str(syndrome)]);
Error = trt(1 + syndrome_de, :);
corrected_code = rem(Error + recd, 2);
disp('Corrected code for second received word:');
disp(corrected_code);

```

Printed for: swarajnangare36@gmail.com
Powered by Octave Online
http://octave-online.net