Exp9.m

```matlab
clc;
clear all;

% Input generator matrix
g = input('Enter the generator matrix: ');
disp('G =');
disp(g);

% Determine the order of the linear block code
[n, k] = size(transpose(g));
disp('The order of the linear block code for given generator matrix is: ');
disp(['n = ', num2str(n), ', k = ', num2str(k)]);

% Generate all possible codewords
u = zeros(2^k, k);
for i = 1:2^k
    for j = k:-1:1
        if rem(i - 1, 2^(-j + k + 1)) >= 2^(-j + k)
            u(i, j) = 1;
        else
            u(i, j) = 0;
        end
    end
end

% Calculate codewords by multiplying message vectors with G
c = rem(u * g, 2);
disp('The possible codewords are: ');
disp(c);

% Calculate minimum Hamming distance
disp('The minimum Hamming distance dmin for given block code = ');
d_min = min(sum(c(2:2^k, :)')); % Exclude the all-zero codeword
disp(d_min);

% Input received codeword
r = input('Enter the received code word: ');

% Calculate the parity-check matrix H
p = g(:, n - k + 1:n);
h = [transpose(p), eye(n - k)];
disp('Parity-check matrix H: ');
disp(h);

% Calculate syndrome of received codeword
ht = transpose(h);
s = rem(r * ht, 2);
disp('Syndrome of the given codeword is: ');
disp(s);

% Error correction based on syndrome
error_pos = -1;
for i = 1:n
    if isequal(ht(i, :), s)
        error_pos = i;
        r(i) = 1 - r(i);  % Flip the bit at the error position
        break;
    end
end

% Display results
if error_pos > 0
    disp(['The error is in bit: ', num2str(error_pos)]);
else
    disp('No single-bit error detected');
end

```

```
disp('The corrected codeword is: ');
disp(r);

```