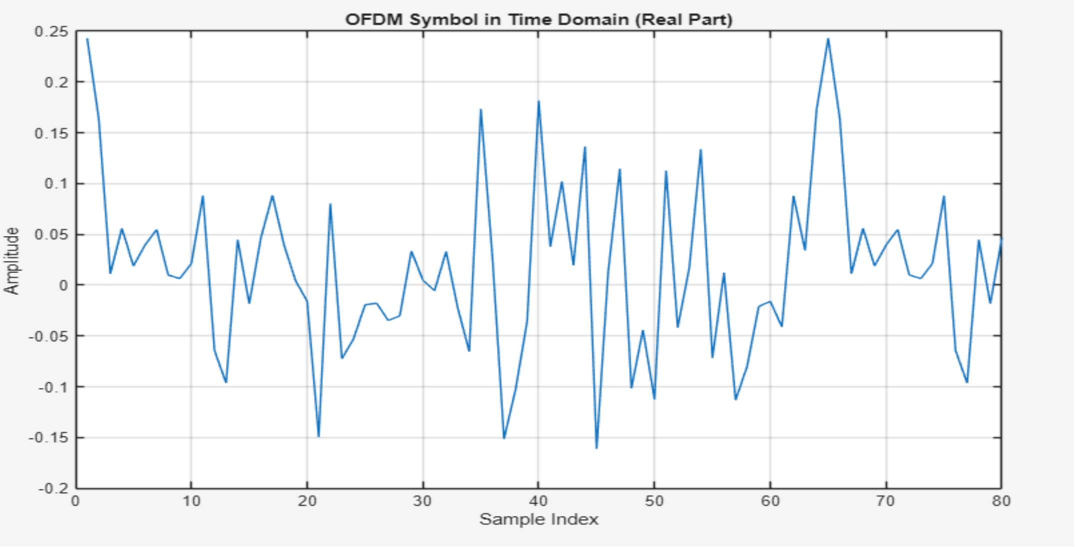
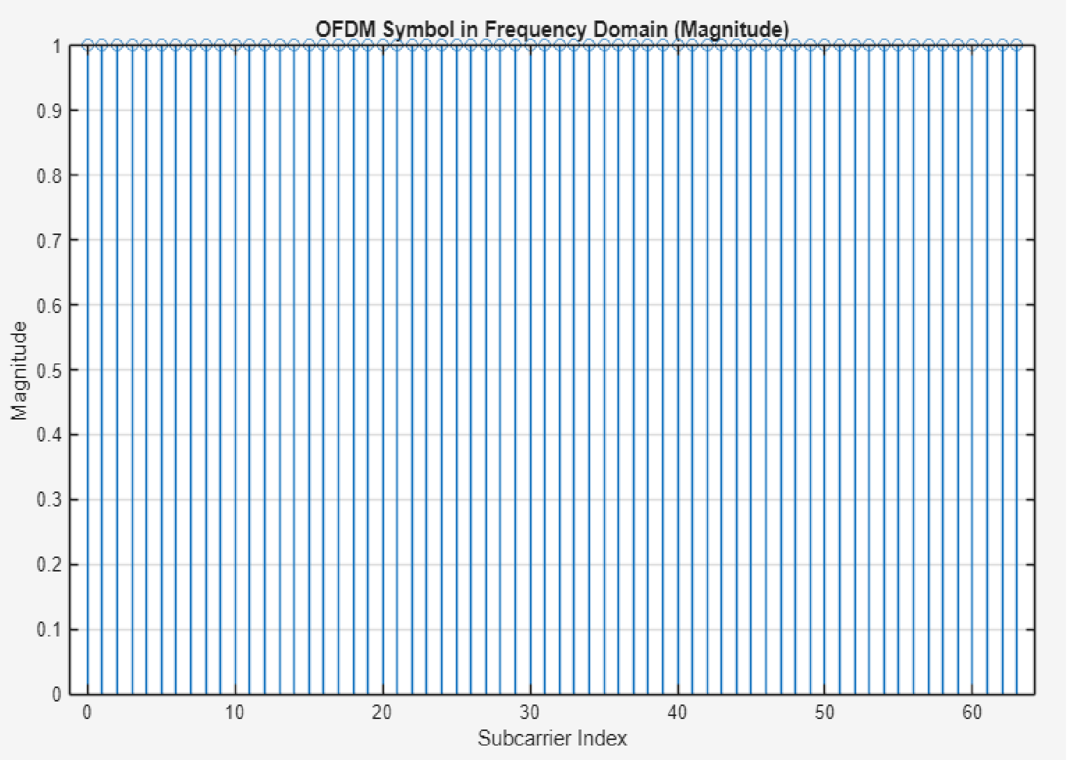
Practical No : 7

```
N = 64;      % Number of subcarriers
CP = 16;     % Length of cyclic prefix
numSym = 5;  % Number of OFDM symbols

% Generate random bits (2 bits per QPSK symbol)
bits = randi([0 1], N*2*numSym, 1);

% QPSK modulation: map bits to symbols
bits_reshape = reshape(bits, 2, []).';
symbols = (1 - 2*bits_reshape(:,1)) + 1j*(1 - 2*bits_reshape(:,2));
symbols = symbols / sqrt(2);  % Normalize power

% Group symbols into OFDM symbols (rows)
ofdm_symbols = reshape(symbols, N, numSym).';

% IFFT to get time domain OFDM signal
time_domain = ifft(ofdm_symbols, N, 2);

% Add cyclic prefix (copy last CP samples to front)
ofdm_with_cp = [time_domain(:, end-CP+1:end) time_domain];

% Received signal (ideal channel, no noise)
received = ofdm_with_cp;

% Remove cyclic prefix
received_no_cp = received(:, CP+1:end);

% FFT to recover frequency domain symbols
received_symbols = fft(received_no_cp, N, 2);

% Flatten for demodulation
received_symbols = received_symbols.';

% QPSK demodulation (decide bits from real and imag parts)
received_bits = zeros(length(bits), 1);
received_bits(1:2:end) = real(received_symbols(:)) < 0;
received_bits(2:2:end) = imag(received_symbols(:)) < 0;

% Calculate errors
numErrors = sum(bits ~= received_bits);
BER = numErrors / length(bits);

fprintf('Bit errors: %d\nBit error rate: %f\n', numErrors, BER);
symbol_idx = 1;

% Frequency domain (QPSK symbols on subcarriers)
figure;
stem(0:N-1, abs(ofdm_symbols(symbol_idx, :)));
```

```
title('OFDM Symbol in Frequency Domain (Magnitude)');
xlabel('Subcarrier Index');
ylabel('Magnitude');
grid on;

% Time domain OFDM symbol (with cyclic prefix)
figure;
plot(real(ofdm_with_cp(symbol_idx, :)));
title('OFDM Symbol in Time Domain (Real Part)');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;
```

OUTPUT:


OFDM Symbol in Time Domain (Real Part)
Amplitude
Sample Index


OFDM Symbol in Frequency Domain (Magnitude)
Magnitude
Subcarrier Index