```
function rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,case_type,fs)
%
% Inputs:
%   rec_sample_seq: The input sample sequence to the channel
%   case_type:      The sampling frequency used to generate the sample sequence
%   fs:             The bit flipping probability
% Outputs:
%   rec_sample_seq: The sequence of sample sequence after passing through the channel
%
% This function takes the sample sequence after passing through the
% channel, and decodes from it the sequence of bits based on the considered
% case and the sampling frequence

if (nargin <= 2)
    fs = 1;
end

No_Bits = length(rec_sample_seq); % number of bits

switch case_type

    case 'part_1'
        %%% WRITE YOUR CODE FOR PART 1 HERE
        % fs >> The bit flipping probability
        No_zeros = round(No_Bits * fs);    % number of ones

        rec_bit_seq = [zeros(1, No_zeros) ones(1, No_Bits-No_zeros)];
        rec_bit_seq = rec_bit_seq(randperm(No_Bits)); % shuffle the vector
        %%%
    case 'part_2'
        %%% WRITE YOUR CODE FOR PART 2 HERE
        % fs >> Number of samples per symbol (bit)
        rec_bit_seq = mode(rec_sample_seq);  % get the most frequent value in each column

        %%%
    case 'part_3'
        %%% WRITE YOUR CODE FOR PART 3 HERE
        rec_bit_seq = rec_sample_seq(1,:);
        %%%
end
```