```
% % Mfile name
%   mtl_aae_sim_float2decusf.m

% Version:
%   Matlab R2007a

% Revised:
%   August 28, 2008

% Authors:
%   Luke Snyder, Dr. Autar Kaw
%   University of South Florida
%   kaw@eng.usf.edu
%   Website: http://numericalmethods.eng.usf.edu

% Purpose
%   To illustrate the concept of the conversion of a floating point binary
%   number to base-10 decimal representation.

% Keywords
%   Binary to Decimal conversion
%   Fixed point register

% Clearing all data, variable names, and files from any other source and
% clearing the command window after each successive run of the program.
clc
clear all

% Inputs:
%    This is the only place in the program where the user makes the changes
%    based on their wishes.
% NOTE: The apostrophe symbol (') must be present around the numbers
% entered into the worksheet or else it will not work!
    % Enter the total number of bits
    tot_bits = '9';

    % Enter the sign of the number: '0' if the number is positive, '1' if
    % the number is negative.
    dec_sign = '0';

    % Enter the sign of the exponent: (same convention applies)
    exp_sign = '0';

    % Enter the mantissa bits
    Mant_bits = '1011';

    % Enter the exponent bits
    Exp_bits = '101';

% **************************************************************************

disp(sprintf('\nConcepts of Conversion of Base-2 Floating Point Binary '))
disp(sprintf('Number to a Base-10 Decimal Number'))
disp(sprintf('\nUniversity of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu'))
disp(sprintf('Website: http://numericalmethods.eng.usf.edu'))
disp(sprintf('\nNOTE: This worksheet illustrates the use of Matlab to convert'))
disp('a base-2 floating point binary number to a base-10 decimal number.')

disp(sprintf('\n***************************Introduction***************************'))

disp(sprintf('\nThe following worksheet illustrates how to convert a base-2 '))
disp(sprintf('floating point binary number to a base-10 decimal number using '))
disp(sprintf('loops and various conditional statements. The user inputs total '))
disp(sprintf('number of bits, decimal sign bit value, exponent sign bit value, '))
disp(sprintf('mantissa bits entry, and exponent bits entry in the Input section of '))
disp(sprintf('the program. The program will then convert the floating point '))
disp(sprintf('binary number into a decimal number.'))

disp(sprintf('\n****************************Input Data******************************'))
fprintf('\n');
strtot = ['     The total number of bits, tot_bits = ',tot_bits];
disp(strtot)
strdec = ['     The sign of the decimal number, dec_sign = ',dec_sign];
disp(strdec)
strexp = ['     The sign of the exponent, exp_sign = ',exp_sign];
disp(strexp)
strMant = ['      The mantissa bits specified, Mant_bits = ',Mant_bits];
disp(strMant)
strExp = ['     The exponent bits specified, Exp_bits = ',Exp_bits];
disp(strExp)

disp(sprintf('\n\n****************************Procedure******************************'))

% We must first check to see if the number of bits entered by the user
% corresponds with the total number of bits specified.
tot_bits = str2num(tot_bits);

% Counting the number of bits entered by the user for each input.
a = length(dec_sign);
b = length(exp_sign);
c = length(Mant_bits);
d = length(Exp_bits);

Totalbits = a + b + c + d;

if tot_bits ~= Totalbits
    disp(sprintf('\nThere is an error in the number of bits entered.  The total'))
    disp(sprintf('number of bits counted does not correspond with the total number'))
    disp(sprintf('of bits entered by the user.  Please make sure the addition of'))
    disp(sprintf('all the bits specified equals the total number of bits specified.'))
else
    disp(sprintf('\nThe total number of counted bits is equal to the number of total'))
    disp(sprintf('number of bits entered by the user.  The calculations will now begin.'))

    % Using if statements to designate the signs of both the decimal number
    % to be approximated and the exponent.
    if str2num(dec_sign) == 0
        Dec_sign = 1;
        Dec_str = 'positive';
    elseif str2num(dec_sign) == 1
        Dec_sign = -1;
        Dec_str = 'negative';
    end

    % Displaying the result
    fprintf('\n');
    str = ['The sign of the decimal number is ',Dec_str,'.'];
    disp(str)

    if str2num(exp_sign) == 0
        Exp_sign = 1;
        Exp_str = 'positive';
    elseif str2num(exp_sign) == 1
        Exp_sign = -1;
        Exp_str = 'negative';
    end

    % Displaying the result
    fprintf('\n')
    str = ['The sign of the exponent is ',Exp_str,'.'];
    disp(str)

    % Approximating the mantissa value as specified by the user.  Note that
    % the length of the array (c) has already been calculated in a previous
    % operation.

    % Here we initialize the "Mant_sum" value at one to account for the "1*2^0"
    % term that exists always in the floating point approximations.
    Mant_sum = 1;

    for i=1:1:c
        % Using the loop to create a number array from the mantissa
        % character array.
        Mant_num(i) = str2num(Mant_bits(i));

        % Summing values of the approximation:
        Mant_sum = Mant_sum + (Mant_num(i) * 2^(-i));
    end

    % Displaying the results
    disp(sprintf('\nThe decimal approximation of the mantissa is, Mant_sum = %g',↙
Mant_sum));

    % Approximating the Exponent value as specified by the user.  Again,
    % the length of the array (d) has already been calculated in a previous
    % operation.
    Exp_sum = 0;

    for i=1:1:d
        % Using the loop to create a number array from the exponent
        % character array.
        Exp_num(i) = str2num(Exp_bits(i));

        % Summing values of the approximation.
        Exp_sum = Exp_sum + (Exp_num(i) * 2^(d-i));
    end

    % Displaying the results
    disp(sprintf('\nThe value for the exponent is, Exp_sum = %g',Exp_sum))

    % Putting all the pieces of the approximation together to form the
    % final base-10 decimal number.
    Final_Dec_Num = Dec_sign  * Mant_sum * 2 ^(Exp_sign * Exp_sum);

    disp(sprintf('\nThe final base-10 value as given by the user specified base-2'));
    disp(sprintf('floating point format is, Final_Dec_Num = %e',Final_Dec_Num));

end


disp(sprintf('\n\n****************************Conclusion******************************'))
disp(sprintf('This worksheet illustrates the use of Matlab to convert a floating '))
disp(sprintf('point binary representation to a base-10 number. Recall that '))
disp(sprintf('floating point representation is used more often than fixed point '))
disp(sprintf('representation due to two primary advantages: floating point '))
disp(sprintf('representation supports a much larger range of values while '))
disp(sprintf('maintaining a relative error of similar magnitude for all numbers.'))


disp(sprintf('\n\n****************************Refrences******************************'))
disp('See: <a href = "http://numericalmethods.eng.usf.↙
edu/mtl/gen/01aae/mtl_gen_aae_txt_floatingpoint.pdf">Floating Point Representation</a>')

disp(sprintf('\n\nLegal Notice: The copyright for this application is owned'))
disp(sprintf('by the author(s). Neither MathWorks nor the author(s)'))
disp(sprintf('are responsible for any errors contained within and are '))
disp(sprintf('not liable for any damages resulting from the use of this'))
disp(sprintf('material. This application is intended for non-commercial,'))
disp(sprintf('non-profit use only. Contact the author for permission if'))
disp(sprintf('you wish to use this application in for-profit activities.'))
```