ECE 540 - Bit Patterns of Floating Point Numbers (FPN) in Underflow, Overflow and Indefinite.

The main program will decrease a FPN until it underflows and also increase a FPN until it overflows, at each step showing the number and its bit pattern. Finally an invalid operation is done and the bit pattern for the invalid is displayed.

The results are condensed and included at the end of this write up. From this output it can be seen that the bit patterns match those described in class.

```
% open file to save results.
fid = fopen( 'PrecisionTest.csv', 'w' );
% Test for valid open
if fid < 1
    disp('File not opened' ); % notify of failure
    return; % exit script
end % end of if statement.

x = 1; % initialize x value.

fprintf( fid, '\nDecrease x\n');

% loop to test values of x < 1.
for k = 1:1076

    fprintf(fid,'%g, %bx\n', x, x ); % print out as number and bits.

    x = x / 2; % reduce x

end % end of for loop.

fprintf( fid, '\nIncrease x\n');
x = 1;

% loop to test values of x > 1.
for k = 1:1076

    fprintf(fid,'%g, %bx\n', x, x ); % print out as number and bits.

    x = x * 2; % increase x

end % end of for loop.

x = NaN; % Set x to indefinite.
fprintf( fid, '\nNot a Number\n%g, %bx\n', x, x ); % print out Not a Number.

fclose( fid ); % close out file.
```

CONDENSED PROGRAM OUTPUT Comments are in black.

```
Decrease x
1, 3ff0000000000000
0.5, 3fe0000000000000
0.25, 3fd0000000000000
0.125, 3fc0000000000000
0.0625, 3fb0000000000000
0.03125, 3fa0000000000000
… lots of numbers.
4.45015e-308, 0020000000000000
2.22507e-308, 0010000000000000
Now going into underflow.
1.11254e-308, 0008000000000000
5.56268e-309, 0004000000000000
Eventually reaching zero
1.97626e-323, 0000000000000004
9.88131e-324, 0000000000000002
4.94066e-324, 0000000000000001
0, 0000000000000000

Increase x
1, 3ff0000000000000
2, 4000000000000000
4, 4010000000000000
8, 4020000000000000
16, 4030000000000000
32, 4040000000000000
… lots of numbers.
Eventually Reaching infinity
4.49423e+307, 7fd0000000000000
8.98847e+307, 7fe0000000000000
Inf, 7ff0000000000000

Not a Number
NaN, fff8000000000000
```