

**Ex:**

Write a Matlab® function called `engnum` that accepts as an argument a real number called `real_number` and does the following:

- i) Checks the value of `real_number` and prints an error message "Err: negative value" and returns if `real_number` is less than zero.
- ii) Checks the size of `real_number` and prints an error message "Err: number < 1e-12" and returns if `real_number` is less than  $10^{-12}$ .
- iii) Checks the size of `real_number` and prints an error message "Err: number > 999e12" if `real_number` is greater than  $999 \cdot 10^{12}$ .
- iv) Prints out the value of `real_number` in engineering format (three digit number between 1 and 999) using prefixes p, n, u (for  $\mu$ ), n, k, M, G, or T. That is, the script prints three digits and the engineering prefix. Some examples of print out:

```
input = 45,800      output = 45.8 k
input = 0.03689     output = 36.9 m
input = 4.033·10-8  output = 40.3 n
input = 6,789,222  output = 6.79 M
```

**SOL'N:**    `function engnum(real_number)`

```
    i)  if real_number < 0
         display('Err: negative value')
         return
    end

    ii) if real_number < 1e-12
         display('Err: number < 1e-12')
         return
    end

    iii) if real_number > 999e12
```

```

        display('Err: number > 999e12')
        return
    end

iv) % Determine power of ten to nearest multiple of three.
eng_pwr_ten = floor(log10(real_number)/3);

% Create string for engineering units.
switch eng_pwr_ten
    case -4
        units_str = 'p'
    case -3
        units_str = 'n'
    case -2
        units_str = 'u'
    case -1
        units_str = 'm'
    case 0
        units_str = ''
    case 1
        units_str = 'k'
    case 2
        units_str = 'M'
    case 3
        units_str = 'G'
    case 4
        units_str = 'T'
    otherwise
        display('Err: failed to find engineering units')
end

% Turn main value into number between 100 and 999 so we get
% 3 digits to print out. 0 means value such as 1.47, 1 means
% value such as 14.7, 2 means value such as 147 (no dec pt).
dec_pt_loc = floor(log10(real_number)) - 3*eng_pwr_ten;
% Print three-digit value as string.
three_dig_str = num2str(dec_pt_loc)

```

```
% Print the three digit number with decimal pt if appropriate.
switch dec_pt_loc
    case 0
        valstr = [three_dig_str(1), '.',three_dig_str(2:3)];
    case 1
        valstr = [three_dig_str(1:2), '.',three_dig_str(3)];
    case 2
        valstr = three_dig_str;
    otherwise
        display('Err: decimal pt location not 0, 1, or 2')
end

% Create string with value, decimal pt, and engineering units.
str = [val_str, ' ',units_str];
display(str)

return
end
```