Table of Contents

..... 1
_____ 2
INITIALIZATION 2
_____ 2
CALCULATIONS 2
_____ 2
COMMAND WINDOW OUTPUT 2
_____ 4
ACADEMIC INTEGRITY STATEMENT 4


```
function [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(n)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This program approximates ln3 by running a sumation to a given
 number of
% terms. It then returns the approximation and the difference between
 it
% and actual ln3.
%
% Function Call
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(n)
%
% Input Arguments
% 1. n % the number of terms with which to run the approximation for
 ln3
%
% Output Arguments
% 1. approxln3 %approximation of ln3 as calculated by the series
% 2. absdiff % the absolute value of the difference between the
% approimation for ln3 and the value given by the log(3) function.
%
% Assignment Information
%   Assignment:       PS 08, Problem 3
%   Team ID:            009-01
%   Paired Partner:  WIlliam, Popovich@purdue.edu
%   Paired Partner:  Ethan Hotson, ehotson@purdue.edu
%   Contributor:        Name, login@purdue [repeat for each]
%   Our contributor(s) helped us:
%     [ ] understand the assignment expectations without
%         telling us how they will approach it.
%     [ ] understand different ways to think about a solution
%         without helping us plan our solution.
%     [ ] think through the meaning of a specific error or
%         bug present in our code without looking at our code.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
%preset outputs to output for invalid input
approxln3 = -99;
absDiff = -99;
```

CALCULATIONS

```
if ~(isscalar(n) && (floor(n) == n) && n > 0) %validate inputs
    fprintf("Error: Invalid n\nn must be a positive, scalar, integer
 value.\n");
else
    approxln3 = 0; %initialize the approximation
    index = 0; %initialize index for while loop

    %compute approximation
    while index <= (n-1)
        newTerm = (1/(4^index)) * (1/(2*index + 1)); %compute nth term
        approxln3 = approxln3 + newTerm; %update the approximation
        index = index + 1; %increment index
    end

    absDiff = abs(log(3) - approxln3); %calculate the absolute
 difference between the approximation and matlab's value
end
```

```
Not enough input arguments.

Error in PS08_ln3_approx_wpopovic_ehotson (line 45)
if ~(isscalar(n) && (floor(n) == n) && n > 0) %validate inputs
```

COMMAND WINDOW OUTPUT

```
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(6)
%
% approxln3 =
%
%     1.0986
%
%
% absDiff =
%
%    2.4006e-05
```

```
%
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson([1,12,3])
% Error: Invalid n
% n must be a positive, scalar, integer value.
%
% approxln3 =
%
%    -99
%
%
% absDiff =
%
%    -99
%
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(0)
% Error: Invalid n
% n must be a positive, scalar, integer value.
%
% approxln3 =
%
%    -99
%
%
% absDiff =
%
%    -99
%
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(0.1)
% Error: Invalid n
% n must be a positive, scalar, integer value.
%
% approxln3 =
%
%    -99
%
%
% absDiff =
%
%    -99
%
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(5)
%
% approxln3 =
%
%     1.0985
%
%
% absDiff =
%
%    1.1278e-04
%
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(10)
%
% approxln3 =
```

```
%
%     1.0986
%
%
% absDiff =
%
%    5.8883e-08
%
% [approxln3,absDiff] = PS08_ln3_approx_wpopovic_ehotson(20)
%
% approxln3 =
%
%     1.0986
%
%
% absDiff =
%
%    2.9754e-14
```

ACADEMIC INTEGRITY STATEMENT

```
% Call your academic integrity statement here
```

Published with MATLAB® R2018b