

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Course Number: ENGR 13300
3  % Semester: Fall 2025
4  %
5  % Description:
6  %     Replace this line with a description of your program.
7  %
8  % Assignment Information:
9  %     Assignment:      14.2.1 MA1 Team 3 (for MATLAB 1 Team task 3)
10 %     Team ID:         007 - 19 (e.g. LC1 - 01; for section LC1, team 01)
11 %     Author:          Mark, Sheng65@purdue.edu
12 %     Date:            e.g. 03/24/2025
13 %
14 % Contributor:
15 %     Name, login@purdue [repeat for each]
16 %
17 %     My contributor(s) helped me:
18 %     [ ] understand the assignment expectations without
19 %         telling me how they will approach it.
20 %     [ ] understand different ways to think about a solution
21 %         without helping me plan my solution.
22 %     [ ] think through the meaning of a specific error or
23 %         bug present in my code without looking at my code.
24 %     Note that if you helped somebody else with their code, you
25 %     have to list that person as a contributor here as well.
26 %
27 % Academic Integrity Statement:
28 %     I have not used source code obtained from any unauthorized
29 %     source, either modified or unmodified; nor have I provided
30 %     another student access to my code. The project I am
31 %     submitting is my own original work.
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33 x = input("Enter the value of x: ");
34 tar = input("Enter the target error threshold percentage: ");
35 fprintf("Target error threshold: %d%%\n", tar, "%");
36 terms = 0;
37
38 approximate = 0;
39 actual = exp(x);
40 diff = 0;
41 error = 100;
42
43 while error > tar
44     approximate = maclaurin(x, terms);
45     diff = approximate - actual;
46     error = abs( diff/actual * 100 );
47     %%fprintf("  debug: errorr = %f\n", error);
48     terms = terms + 1;
49 end
50
51 fprintf("Actual value: %.2f\n", actual);
52 fprintf("Terms needed: %d\n", terms);
53 fprintf("Approximate value: %.2f\n", approximate);
54
55
56 function out = f(n)
57     t = 1;
58     for i = n:-1:1
59         t = t * i;
60     end
61

```

```
62     if n == 0
63         out = 1;
64     else
65         out = t;
66     end
67 end
68
69
70 function out = maclaurin(num, precision)
71     sum = 0;
72     for i = 0:1:precision
73         sum = sum + ( num^i / f(i) );
74     end
75     out = sum;
76 end
```