```
1 % PROBLEM 11 EX 1.2
 2
 3 % FUNCTION DEFN
 5 arc_tan = Q(x) x - (1/3) * x^3 + (1/5) * x^5;
 7 p = pi;
 8
 9 % PART A
10 arc_tan_1 = arc_tan(1/2);
11 arc_tan_2 = arc_tan(\frac{1}{3});
12
13 p_{star} = 4 * (arc_{tan_1} + arc_{tan_2});
15 % ABS ERROR
16
17 abs_error = abs(p - p_star);
18 rel_error = abs_error / pi;
19
20 display(arc_tan_1);
21 display(arc_tan_2);
22
23 display(abs_error);
24 display(rel_error);
25
26 % PART B
27 arc_tan_1_B = arc_tan(1/5);
28 \ arc_{tan_2B} = arc_{tan_{1/239}};
29
30 p_star_B = 16 * arc_tan_1_B - 4 * arc_tan_2_B;
31
32 abs_error_B = abs(p - p_star_B);
33 rel_error_B = abs_error_B / p;
34
35 display(arc tan 1 B);
36 display(arc_tan_2_B);
37
38 display(abs_error_B);
39 display(rel_error_B);
40
41
42
43
44
```

```
1 % FUNCTION THAT FINDS THE ROOT OF THE EQUATION IN AN INTERVAL
  WITH TOLERANCE TOL
3 function rt = bisection_rec(f, a, b, N, TOL)
       % FIND THE MIDPOINT
5
6
       p_n = (a + b) / 2;
7
8
      % CHECK TOLERANCE
       if abs(p_n - b) < TOL
9
10
           rt = p_n;
11
           return;
12
       end
13
14
      % STOPPING CONDITION
15
       if N == 0
16
           rt = p_n;
17
           return;
18
       end
19
20
      % CHECK SIGN
21
       if f(p_n) * f(a) > 0
22
           a = p_n;
23
       else
24
           b = p_n;
25
       end
26
27
      % DO RECURSION ON NEW INPUTS
       rt = bisection_rec(f, a, b, N - 1, TOL);
28
29 end
30
31
32
33
34
35
36
37
```

```
File - /Users/monishwaran/Desktop/math128A/MATLAB_code/HW2/p2_d.m
```

```
1 p = factorial(9);
2 p_star = 3 * sqrt(2 * pi) * (9^9 / exp(9));
3
4 difference = p - p_star;
5 abs_error = abs(difference);
6 relative_error = abs_error / p;
7
8 display(difference)
9 display(abs_error)
10 display(relative_error)
11
```