```
% PROBLEM 11 EX 1.2

% FUNCTION DEFN

arc_tan = @(x) x - (1/3) * x^3 + (1/5) * x^5;

p = pi;

% PART A
arc_tan_1 = arc_tan(1/2);
arc_tan_2 = arc_tan(1/3);

p_star = 4 * (arc_tan_1 + arc_tan_2);

% ABS ERROR

abs_error = abs(p - p_star);
rel_error = abs_error / pi;

display(arc_tan_1);
display(arc_tan_2);

display(abs_error);
display(rel_error);

% PART B
arc_tan_1_B = arc_tan(1/5);
arc_tan_2_B = arc_tan(1/239);

p_star_B = 16 * arc_tan_1_B - 4 * arc_tan_2_B;

abs_error_B = abs(p - p_star_B);
rel_error_B = abs_error_B / p;

display(arc_tan_1_B);
display(arc_tan_2_B);

display(abs_error_B);
display(rel_error_B);
```

```m
% FUNCTION THAT FINDS THE ROOT OF THE EQUATION IN AN INTERVAL
WITH TOLERANCE TOL

function rt = bisection_rec(f, a, b, N, TOL)

    % FIND THE MIDPOINT
    p_n = (a + b) / 2;

    % CHECK TOLERANCE
    if abs(p_n - b) < TOL
        rt = p_n;
        return;
    end

    % STOPPING CONDITION
    if N == 0
        rt = p_n;
        return;
    end

    % CHECK SIGN
    if f(p_n) * f(a) > 0
        a = p_n;
    else
        b = p_n;
    end

    % DO RECURSION ON NEW INPUTS
    rt = bisection_rec(f, a, b, N - 1, TOL);
end
```

```
p = factorial(9);
p_star = 3 * sqrt(2 * pi) * (9^9 / exp(9));

difference = p - p_star;
abs_error = abs(difference);
relative_error = abs_error / p;

display(difference)
display(abs_error)
display(relative_error)

```