

```
% Computes the fixed point of a function using the FPI.
% Written by Zachary Ferguson
```

```
function fixed_point_iteration
    fprintf('Fixed Point Iteration\nWritten by Zachary Ferguson\n\n');

    fprintf('Q1a:\n\tg(x) = (2x+2)^(1/3) = x\n');
    fprintf('\txc = %.10f\n', compute_fixed_point(...
        @(x) (2 * x + 2)^(1 / 3.), 2));

    fprintf('Q1b:\n\tg(x) = ln(7-x) = x\n');
    fprintf('\txc = %.10f\n', compute_fixed_point(@(x) log(7 - x), 2));

    fprintf('Q1c:\n\tg(x) = ln(4-sin(x)) = x\n');
    fprintf('\txc = %.10f\n', ...
        compute_fixed_point(@(x) log(4 - sin(x)), 2));

    A = 3.;
    g = @(x) (x + A / x) / 2.;
    fprintf('Q3a:\n\tg(x) = (x + 3 / x) / 2\n');
    fprintf('\tx0 = 2\n');
    fprintf('\txc = %.10f\n', compute_fixed_point(g, 2));

    A = 5.;
    g = @(x) (x + A / x) / 2.;
    fprintf('Q3b:\n\tg(x) = (x + 5 / x) / 2\n');
    fprintf('\tx0 = 2\n');
    fprintf('\txc = %.10f\n', compute_fixed_point(g, 2));

    fprintf('Q5:\n\tg(x) = cos^2(x)\n');
    xc = compute_fixed_point(@(x) (cos(x))^2, 1, 1e-6);
    fprintf('\txc = %.10f\n', xc);
    fprintf('\n\td/dx g(x) = -2*cos(x)*sin(x)\n');
    fprintf('\t|d/dx g(xc)| = %.10f\n', abs(-2 * cos(xc) * sin(xc)));
    fprintf('\tTherefore g(x) is locally convergent to xc.\n');
end
```

```
% Compute the fixed point of g(x).
function xc = compute_fixed_point(g, x0, tol)
    if nargin < 3
        tol = 1e-9;
    end

    prev_x = x0;
    x = g(x0);
    n = 1;
    while (abs(prev_x - x) > 0.5 * tol)
        prev_x = x;
        x = g(x);
        n = n + 1;
    end
    fprintf('\tn = %d\n', n);
    xc = x;
end
```