Question 1

Implement fixed-point iteration. Your signature should be

function p = fp(g, p0, maxits)

a) g : x → (x + 2/x)/2

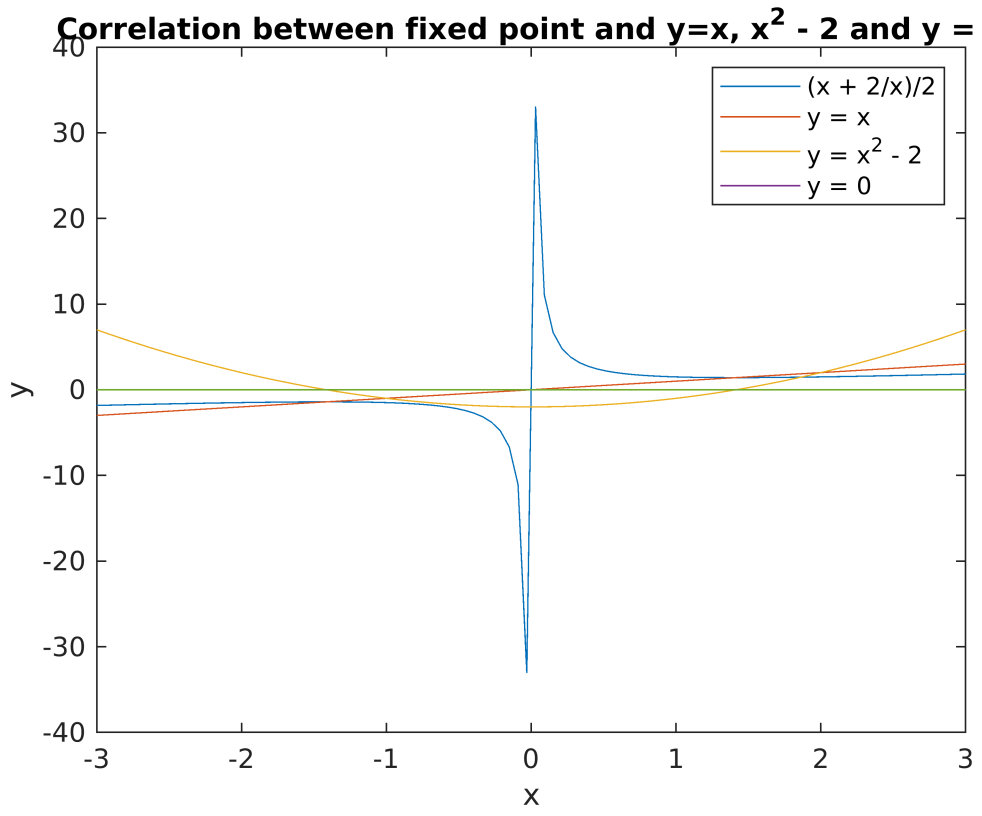
```
g = @(x) (x + 2/x)/2;
fp(g, 1, 3)
```

```
ans =
   1.414215686274510
```

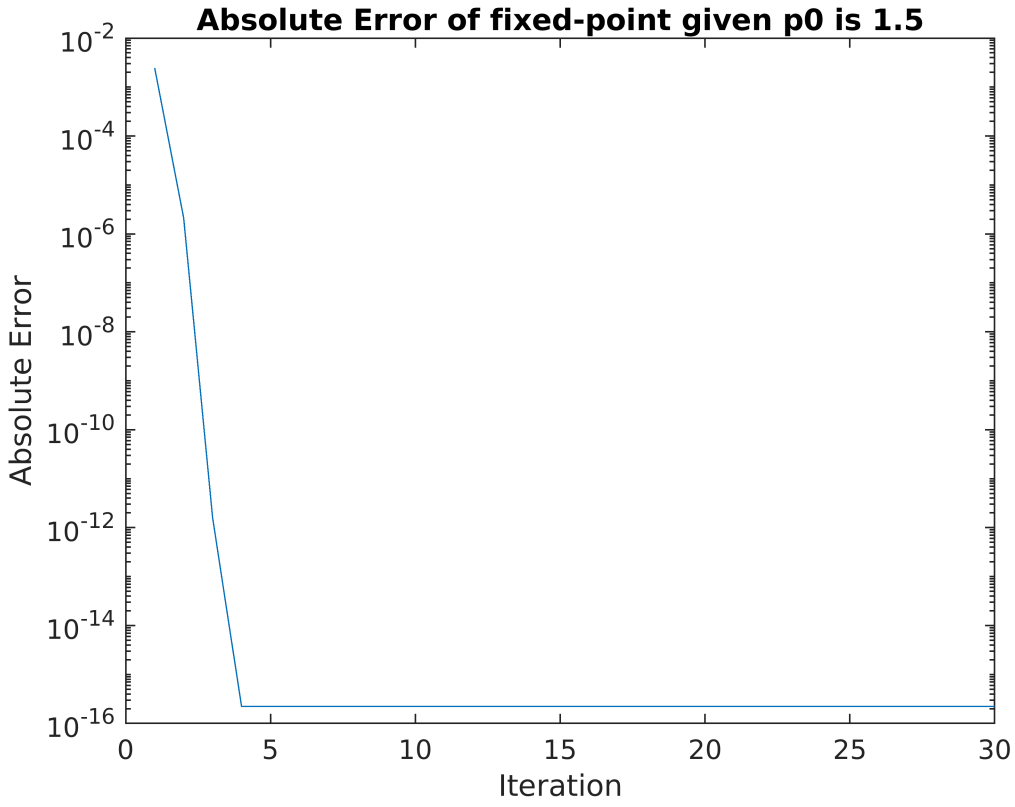
```
x1 = linspace(-3,3,100);
y1 = 0;
for i = 1:length(x1)
    y1(i) = g(x1(i));
end
y2 = x1;

f = @(x) x.^2 - 2;
for i = 1:length(x1)
    y3(i) = f(x1(i));
end
y4 = zeros(100)
```

```
y4 = 100×100
     0     0     0     0     0     0     0     0     0     0     0     0     0 ···
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
     ⋮
```

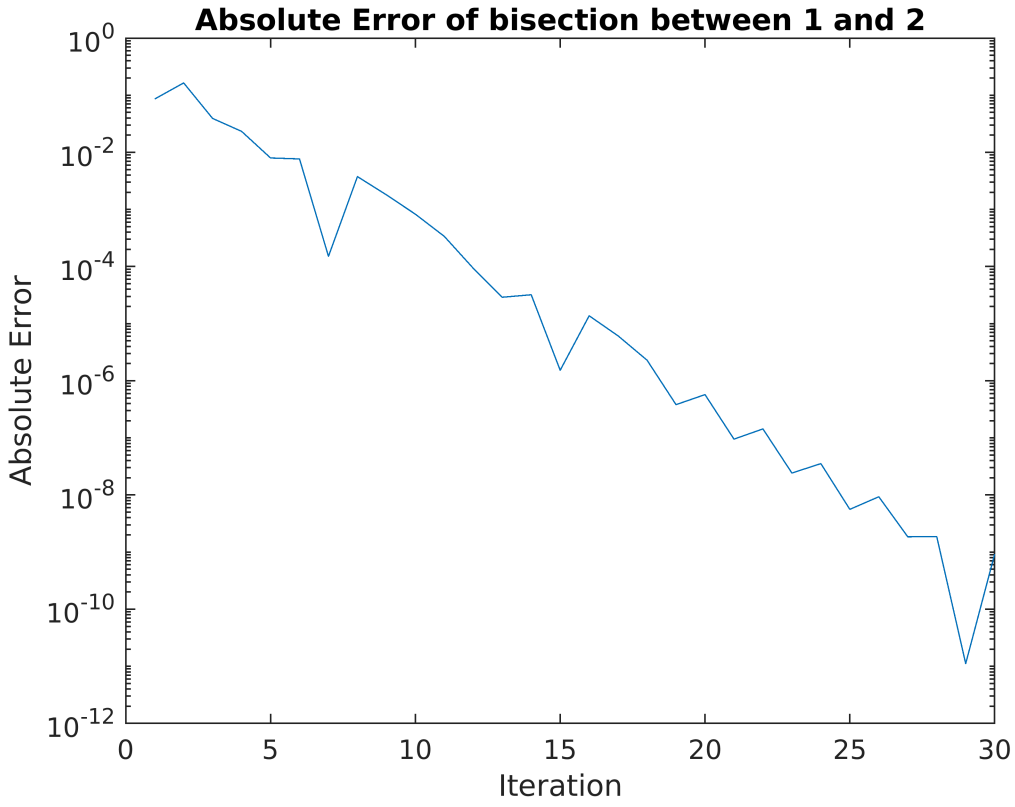
```
plot(x1,y1,x1,y2,x1,y3,x1,y4)
legend('(x + 2/x)/2','y = x', 'y = x^2 - 2', 'y = 0');
title('Correlation between fixed point and y=x, x^2 - 2 and y = 0');
ylabel('y');
xlabel('x');
```




```
format long
[result, y1] = fp(g, 1.5, 30);
[result, y2] = bisection(@(x) x.^2 - 2, 1, 2, eps, 30, 'none');
x1 = linspace(1, 30, 30);
for i = 1:30
    y1(i) = abs(sqrt(2) - y1(i));
    y2(i) = abs(sqrt(2) - y2(i));
end
semilogy(x1, y1)
title('Absolute Error of fixed-point given p0 is 1.5')
xlabel('Iteration')
ylabel('Absolute Error')
```




```
semilogy(x1, y2)
title('Absolute Error of bisection between 1 and 2')
xlabel('Iteration')
ylabel('Absolute Error')
```



i) Done, visually graphed.

ii) if p0 is negative, such as -1, it will approach -sqrt2, if p0 is positive, such as 1, it will approach sqrt2. Maxits can be any value greater than 2 depending on how close p0 is to the actual value.

iii) Drawn.

==========

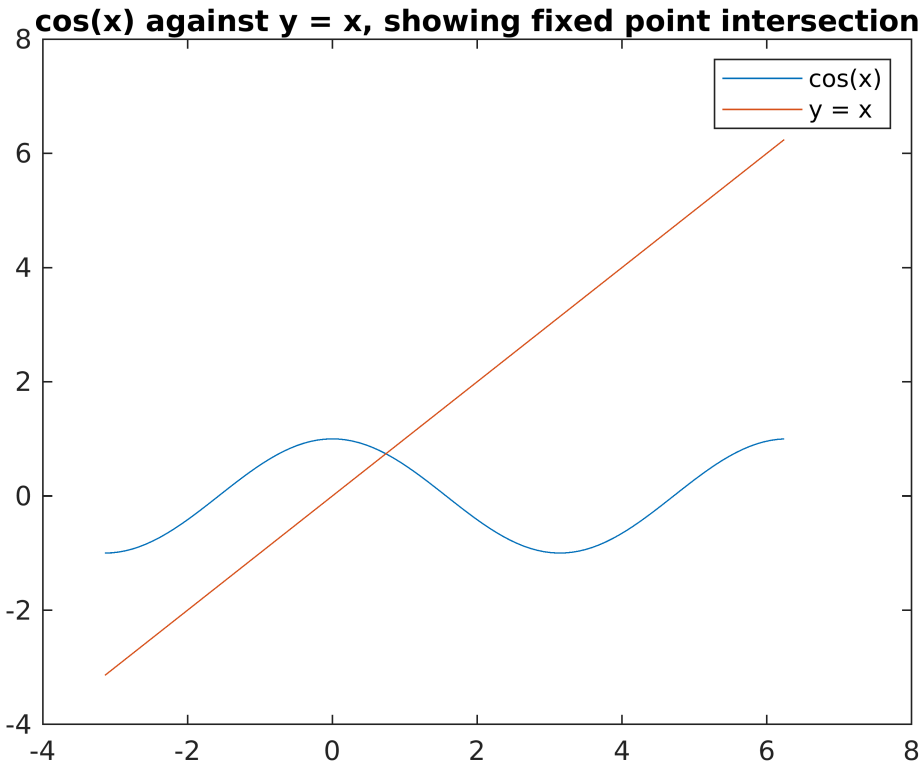
b) g : x → cos(x)

```
g = @(x) cos(x);
[p, y3] = fp(g, 1, 30)
```

```
p =
   0.739087042695332
y3 = 1×30
   0.540302305868140   0.857553215846393   0.654289790497779   0.793480358742566 ···
```

```
x1 = linspace(-3.14,6.24,100);
y1 = 0;
for i = 1:length(x1)
    y1(i) = g(x1(i));
end
y2 = x1;
plot(x1,y1,x1,y2)
legend('cos(x)', 'y = x')
```

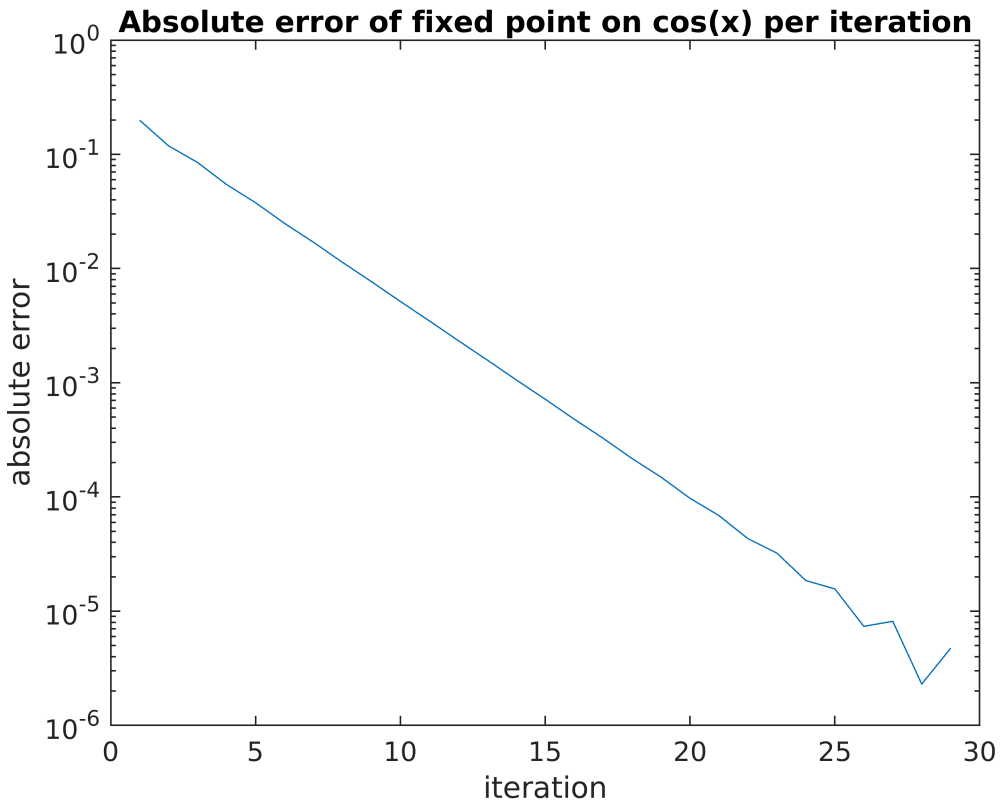
```
title('cos(x) against y = x, showing fixed point intersection')
```




```
x1 = linspace(1,30,30)
```

```
x1 = 1×30
     1     2     3     4     5     6     7     8     9    10    11    12    13 ···
```

```
for i = 1:30
    y3(i) = abs(p - y3(i));
end
semilogy(x1, y3)
title('Absolute error of fixed point on cos(x) per iteration')
xlabel('iteration')
ylabel('absolute error')
```



i) Drawn. Definitely looks to be a fixed point between 0 and 1.

ii) Drawn.

==========

c) g : x → 2x

```
g = @(x) 2 * x;
fp(g, 0, 30)
```

```
ans =
     0
```

p0 will only converge if the starting point is 0. otherwise it expands negatively or positively towards respective infinity.

==========

d) g : x/2 when x != 0

1 when x = 0

```
g = @(x) pwf(x);
[p, y1] = fp(g, 1, 5000)
```

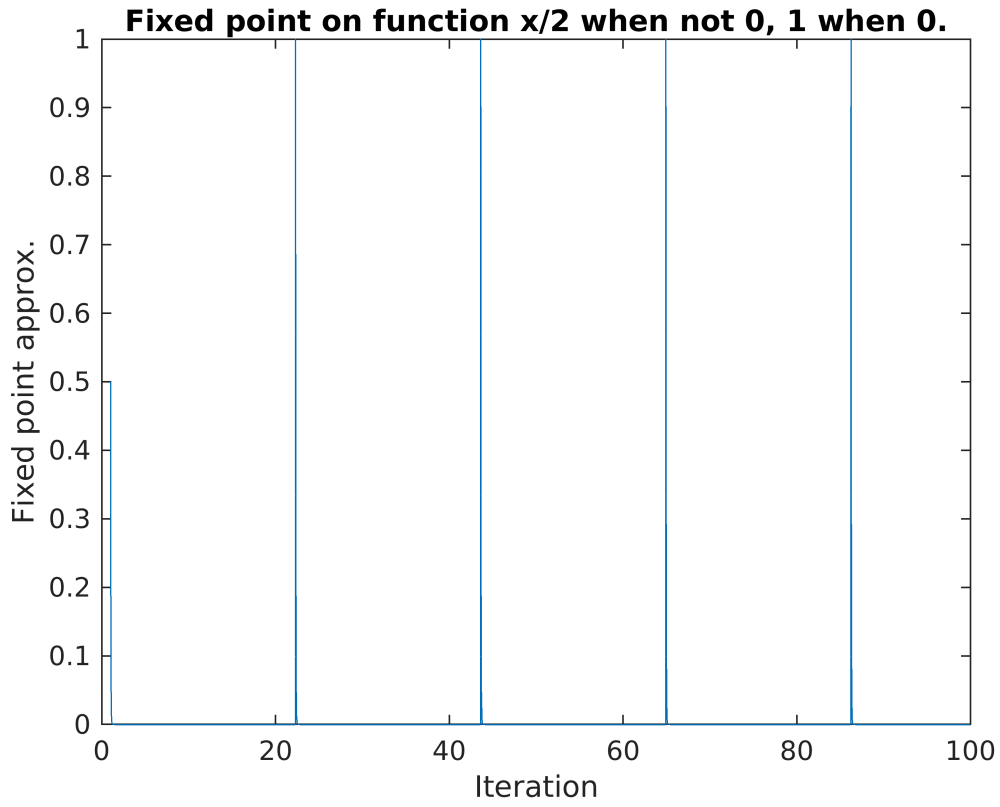
```
p =
    3.041746506072256e-210
```

```
y1 = 1×5000
   0.500000000000000   0.250000000000000   0.125000000000000   0.062500000000000 ···
```

```
x1 = linspace(1,100, 5000)
```

```
x1 = 1×5000
10^2 ×
   0.010000000000000   0.010198039607922   0.010396079215843   0.010594118823765 ···
```

```
plot(x1, y1)
title('Fixed point on function x/2 when not 0, 1 when 0.')
xlabel('Iteration')
ylabel('Fixed point approx.')
```



fp implementation bounces back to 1 every time it comes close to 0 so it is stuck in an infinite loop (or reaches maxiterations).

It would get closer and close to 0 and never reach a finite solution.

```
function [p, y1] = fp(g, p0, maxits)
    p = g(p0);
    y1(1) = p;
    for i = 2:maxits
        p = g(p);
        y1(i) = p;
    end
```

```
end

function value = pwf(x)
    if x == 0
        value = 1;
    else
        value = x / 2;
    end
end

function [r, y] = bisection(f, a, b, tol, maxits, mode)
    x = 0;
    y = 0;
    format long
    for i = 1:maxits
        r = ( a + b ) / 2;
        if (mode == 'iter')
            fprintf('Iteration: %d Value: %.5f\n', i, r)
        end

        x(i) = i;
        y(i) = r;

        if (sign(f(r)) == sign(f(a)))
            a = r;
        else
            b = r;
        end

        if (abs(a - b)<tol)
            break
        end

    end

    if (mode == 'plot')
        title('Value of Bisection at each Iteration')
        scatter(x, y)
    end
end
```