

```
function [lambda, eig, err] = powerEig(A, x, TOL)

% Define initial values and prepare for iteration
[x_p, p] = max(abs(x));
x = x/x_p;

lambda = 0;
eig = zeros(length(x), 1);
err = 0;

for i = 1:1000

    % Begin by setting the y vector equal to the product of A and x,
    % then use y to define the eigenvalue
    y = A*x;
    lambda(i) = y(p);
    [y_p, p] = max(abs(y));

    % If the infinity norm of y is 0, then the function found the zero
    % eigenvalue of A. Choose a different initial x vector and start
    % over.
    if (y_p == 0)
        eig = x;
        disp("A has the eigenvalue 0, select a new vector and restart")
        return
    end

    % Update the error and the x vector for the next iteration.
    err(i) = norm(x - (y/y(p)), inf);
    x = y/y(p);
    eig(:, i) = x;

    % If the calculated error is less than the given tolerance, output
    % x as the dominant eigenvalue.
    if (err(i) < TOL)
        return
    end

end

disp("Need more iterations")

end
```