Problem E9.10

```
clear vars
syms x1 x2 real
X = [x1; x2];
f = (x1 + x2)^4 - 12*x1*x2 + x1 + x2 + 1;
g = gradient(f, X);
h = hessian(f, X);
initial_guesses= [-1 -1; 0 0; 1 1];
max_iters = 10;
epsilon = 1e-5;
```

Steepest Descent:

$x_{k+1} = x_k - \alpha\, g_k, \alpha = 0.03$

```
alpha = 0.03;
for i=1:length(initial_guesses)
    current_x = initial_guesses(i,:)';
    reached = 0;
    iters = 0;
    while iters < max_iters
        next_x = current_x - alpha*subs(g, X, current_x);
        iters = iters + 1;
        if abs(subs(g, X, current_x)) < epsilon
            fprintf('SD terminated at [%f; %f] with initial guess [%f; %f] (%d/%d iters)\n\n',
                next_x(1), next_x(2), initial_guesses(i,1), initial_guesses(i,2), iters, max_it
            reached = 1;
            break;
        end
        current_x = next_x;
    end
    if ~reached
        fprintf('SD terminated at [%f; %f] with initial guess [%f; %f] (%d/%d iters)\n\n', ...
            next_x(1), next_x(2), initial_guesses(i,1), initial_guesses(i,2), iters, max_iters
    end
end
```

```
SD terminated at [-0.650420; -0.650420] with initial guess [-1.000000; -1.000000] (10/10 iters)

SD terminated at [-0.648517; -0.648517] with initial guess [0.000000; 0.000000] (10/10 iters)

SD terminated at [0.564907; 0.564907] with initial guess [1.000000; 1.000000] (10/10 iters)
```

We can see that all of the initial guesses have stepped downhill towards either of the strong minimums and none of them resulted in the algorithm getting stuck in the saddle point.

Newton's Method:

$x_{k+1} = x_k - A_k^{-1} g_k, A_k \equiv \nabla^2 F(x)_{x=x_k}$

```
for i=1:length(initial_guesses)
    current_x = initial_guesses(i,:)';
```

```
    reached = 0;
    iters = 0;
    while iters < max_iters
        next_x = current_x - inv(subs(h, X, current_x))*subs(g, X, current_x);
        iters = iters + 1;
        if abs(subs(g, X, current_x)) < epsilon
            fprintf('NM terminated at [%f; %f] with initial guess [%f; %f] (%d/%d iters)\n\n',
                next_x(1), next_x(2), initial_guesses(i,1), initial_guesses(i,2), iters, max_it
            reached = 1;
            break;
        end
        current_x = next_x;
    end
    if ~reached
        fprintf('NM terminated at [%f; %f] with initial guess [%f; %f] (%d/%d iters)\n\n', ...
            next_x(1), next_x(2), initial_guesses(i,1), initial_guesses(i,2), iters, max_iters)
    end
end
```

```
NM terminated at [-0.650420; -0.650420] with initial guess [-1.000000; -1.000000] (6/10 iters)

NM terminated at [0.084969; 0.084969] with initial guess [0.000000; 0.000000] (4/10 iters)

NM terminated at [0.565451; 0.565451] with initial guess [1.000000; 1.000000] (6/10 iters)
```

Convergence occurs much faster with Newton's Method but ws we can see, [0; 0] gets stuck at the Saddle Point, as opposed to SD which is very unlikely to get stuck in a Saddle Point. Additionally, we know that the performance of Newton's Method depends very highly on the initial guess and is, therefore, slightly unreliable in that regard.