HW 2

1: Building blocks Let $d_1, d_2, u, v \in \mathbb{R}^n$ be vectors and define matrices $D_1 = \text{diag}(d_1)$, $D_2 = \text{diag}(d_1)$, and $L = uv^T$. Write an efficient MATLAB or Julia code ($O(n)$ time) that solves the system

$$\begin{bmatrix} D_1 & L \\ 0 & D_2 \end{bmatrix} x = c$$

The function signature should look like

```
% MATLAB/Octave version
function x = hw2solve(d1, d2, u, v, c)

% Julia version (returns a vector x)
function hw2solve(d1, d2, u, v, c)
```

2: Pi, see! The following routine estimates π by recursively computing the semiperimeter of a sequence of 2^{k+1}-gons embedded in the unit circle:

```
N = 4;
L(1) = sqrt(2);
s(1) = N*L(1)/2;
for k = 1:30
  N = N*2;
  L(k+1) = sqrt( 2*(1-sqrt(1-L(k)^2/4)) );
  s(k+1) = N*L(k+1)/2;
end

semilogy(1:length(s), abs(s-pi));
ylabel('|s_k-\pi|');
xlabel('k')
```

Plot the absolute error $|s_k - \pi|$ against k on a semilog plot. Explain why the algorithm behaves as it does, and describe a reformulation of the algorithm that does not suffer from this problem.

3: Low rank limbo Suppose $u, v \in \mathbb{R}^n$, and let $L = uv^*$. Show that

- $\|L\|_1 = \|u\|_1 \|v\|_\infty$
- $\|L\|_\infty = \|u\|_\infty \|v\|_1$
- $\|L\|_F = \|u\|_2 \|v\|_2$