Solutions to Homework 3

Help Center

Problem odd_index:

```
function out = odd_index(M)

    out = M(1:2:end, 1:2:end);
end
```

Problem int_col:

```
function v = int_col(n)

    v = [n 1:n-1]';
end
```

Note that this is just one possible solution. There are many others.

Problem rich:

```
function usd = rich(cent)

    usd = [0.01 0.05 0.10 0.25] * cent';
end
```

We use the fact that matrix multiplication sums up a set of products. Multiplying a row vector with a column vector will result in a scalar. Here it performs the exact calculations we need.

Problem light_time:

```
function [mins km] = light_time(mile)

    km = mile * 1.609;
    mins =  km / 3e5 / 60;
end
```

Problem pitty:

```
function c = pitty(ab)

    c = sqrt(ab(:,1) .^ 2 + ab(:,2) .^2);
end
```

Problem pitty (alternative solution):

```
function c = pitty(ab)
```

```
    c = sqrt(sum(ab' .^ 2))';
end
```

Here we use the fact that the function sum works column by column. So, transposing and then squaring every element will put the squares of the corresponding a-s and b-s into columns. The function sum then adds them up, and sqrt computes each element's square root. Finally, we need to transpose the result back into a column vector.

Problem bottom_left:

```
function M = bottom_left(N,n)

    M = N(end-n+1:end, 1:n);
end
```

We need the last n rows and the first n columns. The only trick here is that we need end-n+1, because end-n:end would get us n+1 indexes and not n as required.

Problem mean_squares:

```
function mm = mean_squares(nn)

    mm = mean((1:nn).^2);
end
```

Problem hulk:

```
function H = hulk(v)

    H = [v' (v').^2 (v').^3];
end
```

Here we need to remember to transpose the vector to get the required arrangement.

Created Mon 11 May 2015 8:00 PM PDT

Last Modified Tue 3 Nov 2015 1:08 PM PST