CS 100M: Prelim 2 Review Questions

1. (Lec 7) Change this function so that the initial guess and `nSteps` are input parameters.

```
  function y = MySqrt1(x)
% Post: y is the square root of x
% Pre:  x is a real number that satisfies 1 <= x <= 4

nSteps = 4;
y = 1 + (x-1)/3;
for i=1:nSteps
    y = (y + x/y)/2;
end
```

Call your function `MyNewSqrt(x,nSteps,InitialGuess)`. Write a script that prints the value returned by `MyNewSqrt` when it is applied to estimate $\sqrt{2}$ for the 12 input choices given in this table:

`nSteps:`	2	2	2	2	3	3	3	3	4	4	4	4
`InitialGuess:`	1.0	1.2	1.4	1.6	1.0	1.2	1.4	1.6	1.0	1.2	1.4	1.6

Hint: There will be a nested loop.

2. (Lec 7) Write a script that assigns to `n` the largest value that can be returned by

```
  function m = UpDown(k)
% Post: m is the number of steps required for the UpDown sequence to reach one.
% Pre:  k is any positive integer.
```

assuming that the value of `k` is 100 or less. I.e., find the maximum value of `UpDown(1)`, `UpDown(2)`,...,`UpDown(100)`.

3. (Lec 8) Write a script that plays one million games of First-to-20 and prints the number of slaughters. A slaughter occurs when the winner wins by more than 10. Recall the how the game is played/simulated:

```
  function [nHeads,nTails] = FirstTo(N)
% Post
%   Simulates a game of First To N.
%   nHeads is the final score of player H.
%   nTails is the final score of player T.
%
% Pre
%   N is a positive integer
    nHeads = 0; nTails = 0;
    while (nHeads < N) && (nTails < N)
       x = rand;
       if x<=.5
          nHeads = nHeads + 1;  % Heads
       else
          nTails = nTails + 1;  % Tails
       end
    end
```

4. (Lec 8) Assume that the functions `F` and `G` are as follows

```
  function  x = F(y,z)
     x = 2*y + z;

  function y = G(z)
      x = 2*z;
      y = F(z,x);
```

What value is assigned to `z` by the script

```
   y = 3;
   z = G(y);
```

5. (Lec 9) Describe the output of

```
x = zeros(1,5);
fork=1:5
   x(k) = 10*k;
end
for k=1:5
   x(k) = x(6-k);
   fprintf('%d  %d',k,x(k))
end
```

6. (Lec 10) Modify the following function so that walk terminates after `n` steps or when the "walker" is more than a distance of 10 units from `(x0,y0)`, whichever comes first.

```
function randomWalk(n,x0,y0)
% Post: make n steps of random walk starting from (x0,y0), show path
% Pre:  x0,y0 are real, n>0

% possible movements:  ( deltaX(i), deltaY(i) )
deltaX= [ 1 -1  0  0];
deltaY= [ 0  0  1 -1];

x= [x0 zeros(1,n)];  % trajectory in x direction
y= [y0 zeros(1,n)];  % trajectory in y direction

% Perform walk, each step is based on a random integer
for k = 2:n+1
    r= ceil(4*rand(1,1));  % get a random integer in range 1..4
    x(k)= x(k-1) + deltaX(r);
    y(k)= y(k-1) + deltaY(r);
end

% Show the walk
plot(x,y,x(1),y(1),'r*',x(end),y(end),'ro')
axis('equal')
title([num2str(n) ' steps of random walk from * to o'])
```

7. (Lec 11) Complete the following function so that it performs as specified.

```
  function P = Perimeter(x,y)
% Post:
%   P is the perimeter of the polygon
% Pre:
%   x and y are row n-vectors that define a polygon
%   having vertices (x(i),y(i)), i=1:n
%
```

8. (Lec 11) Suppose `x = rand(1,1000)`. Write a script that assigns to a variable `n` the number of values in `x` that are strictly bigger than the average of all the values in `x`. Hint. You need a loop to compute the average and then a second loop to compute complete the problem.

9. (Lec 12) Write a function `c = BridgeCost(b,h,n)` that the returns the sum line segment lengths in the plot window when

```
  function DrawBridge(x,y,b,h,n)
% Post:
%   Draws a simple truss with n sections
%   Each section has base b and height h.
%   The bottom left of  the truss is situated at (x,y)
% Pre:
%   x and y are real numbers
%   b and h are positive real numbers
%   hold is on
```

is called. The bridge sections are specified by

```
  function DrawTriangle(x,y,b,h)
% Post:
%   Draws in the current figure window a triangle with
%   vertices (x,y), (x+b,y), and (x+b/2,y+h). The median
%   connecting (x+b/2,y) and (x+b/2,y+h) is also displayed.
%   hold is on
% Pre:
%   x,y,b, and h have real values.
%   hold is on
```

10. (Lec 13) Complete the following function so that it performs as follows

```
   function k = minIndex(x)
% Post: k is the index of the component of x that has the minimum value. i.e., x(k) is
%           is the smallest of the values x(1), x(2),...,x(n-1),x(n)
% Pre:  x is a 1-by-n array of real numbers
```

11. (Lec 13) Complete the following function so that it performs as specified:

```
  function who = Cheapest(C)
% Post:
%   who is a 1-by-m array. For i=1:m, who(i) is the index of the factory that can
%        most cheaply manufacture product i. (Do not worry about ties.)
% Pre:
%   C is an m-by-n "cost matrix". For p=1:m and f=1:n,
%        C(p,f) is what it costs factory f to make
%        product p.
```

12. (Lec 13) Suppose `A` is a matrix with positive entries. Write a script that prints the largest value of each row.

13. (Lec 14) Complete the following function so that it performs as specified

```
  function B = NewLR_Flip(A)
% Post:
%   B is an m-by-n matrix.
%   The left half of B is the same as the right half of A.
%   The right half of B is the same as the left half of A.
% Pre:
%   A is an m-by-n matrix with n even
```