Question 1: (10 points)

(a) Write in the box below the first **six** lines of output that would be produced by executing the following script. Write on the blank the total number of lines of output that would be produced by the script.

```
x = [10 20 30 40 50 60];
z = [ 3  1  2  1  4  1];
for a = 1:6
    for b = 1:z(a)
        disp(b)
    end
    disp(x(a))
end
```

Solution:

```
     1
     2
     3
    10
     1
    20
```

How many lines of output will be produced in total? $\boxed{\textbf{18}}$

(b) What will be printed when the following script is executed? Use the specified print format.

Script	*Function*
`a = 8;`	`function b = woo(a)`
`b = 4;`	
`c = woo(a);`	`b = 1;`
`fprintf('a is %d\n', a)`	`a = a - b;`
`fprintf('b is %d\n', b)`	
`fprintf('c is %d\n', c)`	`fprintf('a is %d\n', a)`

Solution:

```
a is 7
a is 8
b is 4
c is 1
```

Question 2: (15 points)

(a) Fill in the blanks below so that the comment is appropriate and the code fragment would display the values of the even-indexed components of vector y, i.e., y(2), y(4), y(6), ...

```
    y= rand(1, floor(rand*100)+2);  % The length of y is at least ____ 2 ______

    % Example solution 1

    for k = 2 : 2 : length(y)
        disp(  y(k)  )
    end

    % Example solution 2

    for k= 1:length(y)/2
        disp(  y(2*k)  )
    end
```

(b) Implement the following function as specified:

```
  function s = smoothVector(v)
  % Smooth vector v by averaging adjacent values
  % v:  a vector of real values.  Assume that the length of v is > 2.
  % s:  the smoothed vector that is one component shorter than v.
  % For example, if  v=[-2  6  3  4  8.2]  then  s=[2  4.5  3.5  6.1]
  % Do NOT use any built-in functions other than length and zeros.
```

Example solution:

```
s= zeros(1, length(v)-1);  % initialization not necessary

for k= 2:length(v)

    s(k-1)= (v(k-1) + v(k))/2;
end
```

Question 3: (25 points)

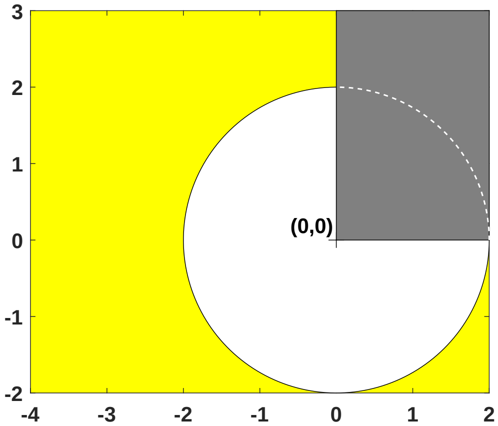
(a) Implement the following function as specified:

```
function [x,y] = randPos(a,b,c,d)
% Generate a random position (x,y) in Cartesian coordinates.
% a,b,c,d are numeric scalars. a<b and c<d
% x is uniformly random in the open interval a to b
% y is uniformly random in the open interval c to d
```

Example solution:

```
x = rand*(b-a)+a;
y = rand*(d-c)+c;
```

(b) A certain script produces the figure shown on the right. Write the code that you would add to such a script in order to plot one asterisk at a randomly generated location within the axes area. The asterisk should be red if it is in the yellow area of the figure, blue if it is in the white part of the circle, or yellow if it is in the gray rectangle.



The white disk has radius 2 and is centered at (0,0). The gray rectangle has a width of 2 and a height of 3 and its bottom-left corner is at (0,0). The x-coordinate of the generated point should be equally likely to be anywhere from -4 to 2 while the y-coordinate should be equally likely to be anywhere from -2 to 3, excluding the boundaries. Assume that the point is not on any edge of the disk or the black rectangle.

Plot *only one* asterisk. Assume `hold` is `on`. Make effective use of function `randPos` (assume it has been implemented correctly and is accessible).

Reminders: The distance between two points (x_1, y_1) and (x_2, y_2) is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. The command to plot a red asterisk at (1,2) is `plot(1,2,'r*')`.

Example solution:

```
[x, y] = randPos(-4,2, -2,3);

if  x>0 && y>0  % in gray rectangle

    plot(x,y,'y*')
else            % outside gray rectangle

    d= sqrt(x^2 + y^2);  % dist from point to disk ctr

    if  d < 2   % in disk

        plot(x,y,'b*')
    else        % outside disk

        plot(x,y,'r*')
    end
end
```

Question 4: (25 points)

Implement the following function as specified:

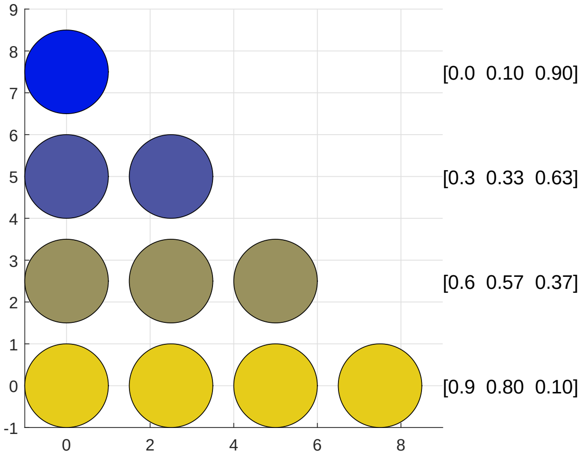
```
function [remain, nTurns, bestWinStreak]=gamblersRuin(d)
% Simulate a game in which a gambler starts with d dollars.  On each turn
%   the gambler is equally likely to win $1 or lose $1.  The gambler leaves
%   and the simulation ends when he has at least doubled his money, when he
%   has less than $1, or after he has played 100 turns, whichever occurs first.
% d: the dollar amount the gambler starts with.  Assume d > 1
% remain: the dollar amount the gambler leaves with
% nTurns: the number of turns the gambler plays in the simulation
% bestWinStreak: the longest consecutive run of $1 gains during the
%   simulation.  If the gambler never wins a turn then bestWinStreak is 0.
% For example, if the simulation has this sequence of outcome:
%   win loss loss win win win loss loss win win loss loss loss loss loss
% then the gambler's longest winning streak is 3.
```

Example solution:

```
t= 0;              % #turns so far
bestWinStreak= 0;  % best winning streak so far
winStreak= 0;      % length of current winning streak
goal= 2*d;

while d<goal && d>=1 && t<100

    t= t + 1;

    if rand < .5  % win

        d= d + 1;

        winStreak= winStreak + 1;

        if winStreak > bestWinStreak

            bestWinStreak= winStreak;
        end
    else          % loss
        d= d - 1;

        winStreak= 0;
    end
end
remain= d;
nTurns= t;
```

Hint: DECOMPOSE! Observe that the winning streak does not control (stop) the simulation. So first work on the simulation without calculating the winning streak; afterwards revise your code to include the winning streak.

Question 5: (25 points)



Complete the function below as specified. *Do not* use any built-in functions other than `length` and `zeros`. The diagram on the right shows an example graphic produced by the following statements:

```
    blu=[0 .1 .9]; yel=[.9 .8 .1];
    diskTri(4, .5, blu, yel)
```

Assume the availability of the function `DrawDisk`. For example, the command

```
    DrawDisk(3, 2, .5, [1 1 0])
```

draws a yellow disk of radius 0.5 centered at (3,2). Your code draws only the disks. The grid lines and the rgb values are shown for your convenience; do not draw them.

```
function diskTri(n, s, cTop, cBot)
% Draw a triangle of disks; there are n disks on each side of the triangle
%   where n is an integer greater than 2.
% The center of the lower left disk is at (0,0).  The disk centers line up
%   vertically and horizontally.  The disks have unit radius.
% s is the minimum distance between adjacent disks where s>0.
% cTop and cBot are vectors of rgb values.  The single disk at the top left has
%   the color cTop; the disks in the bottom row have the color cBot; the rows of
%   disks in between vary uniformly in color (linearly interpolated).

close all; figure; axis equal; hold on
```

Example solution:

```
d= 2+s;       % distance from center to center

for r= n:-1:1  % Go from bottom row (row n) to top row (row 1)

    % y-coord of rth row
    y= (n-r)*d;

    % Color for rth row
    f= (r-1)/(n-1);
    colr= f*cBot + (1-f)*cTop;

    % At rth row, draw r disks starting from the left
    for c= 1:r

        x= (c-1)*d;
        DrawDisk(x,y,1,colr)
    end
end



hold off
```

Hint: DECOMPOSE! First work on drawing the disks at the correct locations all in one color; then revise your code to deal with the color interpolation.