

Question 1: (25 points)**Part (a): (3 points)**

What are the final values of variables x and y?

```
x= 2;
y= 7*x;
x= y;
```

x

14

y

14

Part (b): (3 points)

What are the final values of variables x and y?

```
x= 5;  y= 6;
if x<y
    x= y;  y= x;
end
if x>y
    x= 5;  y= 6;
end
```

x

6

y

6

Part (c): (3 points)

What is the output produced by this script?

```
x= 12;
for k= 100:102
    x= x/2;
    y= k-x
end
```

```
y = 94
y = 98
y = 100.5
```

(Just writing the values is fine)

Question 1, continued**Part (d):** (8 points)

Assume that x is a variable whose value is larger than one. Fill in the loop condition so that the following fragment prints the smallest integer strictly greater than x whose square root is an integer.

```
k= ceil(x);
while _____
    k= k+1;
end

disp(k)
```

Some possibilities:

```
k==x || sqrt(k)~=floor(sqrt(k))
~( k>x && sqrt(k)==floor(sqrt(k)) )
```

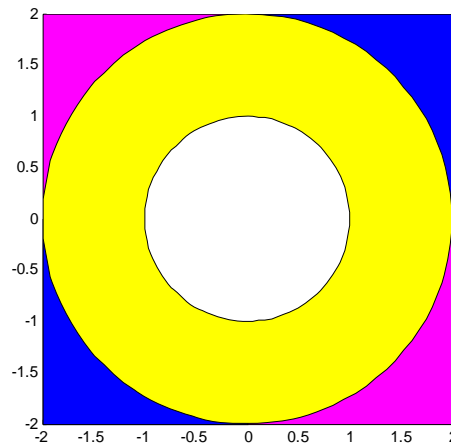
Part (e): (8 points)

Assume that a , b , c , and d are initialized variables whose values satisfy $a < b$ and $c < d$. Fill in the condition so that the message is printed if the intervals $[a,b]$ and $[c,d]$ do not intersect.

```
if _____ b < c || d < a _____
    disp('[a,b] and [c,d] do not intersect')
end
```

Question 2: (25 points)

The 4-by-4 square target to the right is centered at (0,0) and has six regions. It has a white inner circle with unit radius and a yellow ring with outer radius 2. The upper right and lower left corner regions are colored blue. The lower right and upper left corner regions are colored pink. Complete the script below so that it generates a random point within the target and prints the name of the color of the region where it is located. (E.g., `disp('yellow')`). You may assume that the point is never on the boundary of a region.



```

% Code to set up the window, draw the squares and circles ...
% Code not shown. Assume it is correct.

% x, y are randomly generated in interval (-2,2). Do not draw the point.

x = -2+4*rand;
y = -2+4*rand;

% Point (x,y) lies in which region?

r = sqrt(x^2 + y^2);

if r <= 1
    disp('white')

elseif r<=2
    disp('yellow')

else
    if x>0 && y>0 || x<0 && y<0
        disp('blue')
    else
        disp('pink')
    end
end

% A less efficient solution would just have a bunch of ifs lined up...
% OK this time
r = sqrt(x^2+y^2)
if r<=1
    disp('white')
end
if 1<=r && r<=2
    disp('yellow')
end
if r>2 && x>0 && y >0
    disp('blue')
end
% ... etc

```

Question 3: (25 points)

The value of $\frac{\pi^2}{8}$ may be approximated by the series

$$1 + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots$$

Write a script that evaluates this expression, ignoring all terms that are strictly smaller than .000001. Your script should display the number of terms summed and the sum.

```

k= 0;           % number of terms summed so far

s= 0;          % sum so far

term= 1;

while term >= .000001

    s = s + term;

    k = k+1;

    term = 1/(2*k+1)^2;
end

disp(sprintf('Sum is %f', s))           % Printing just the value is ok
disp(sprintf('%d terms summed', k))

% Alternate solution
term= 1; % value of current term

k= 1; % current term number

s= 0; % sum so far

while term >= 0.000001

    s= s + term;

    k= k+1;

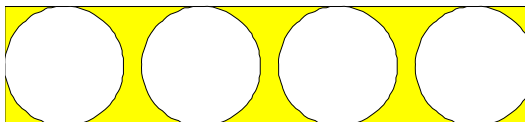
    term= 1/(2*k-1)^2;

end
disp(sprintf('Sum is %f', s))           % printing just the value is ok
disp(sprintf('%d terms summed', k-1))

```

Question 4: (25 points)

Assume that L and W are initialized variables whose values satisfy $0 < W < L/2$. Write a script that draws an L -by- W rectangle and as many non-overlapping disks as possible within the rectangle. The diameter of each disk is the width of the rectangle. The disks are evenly distributed within the rectangle with the leftmost and rightmost disks tangent to the left and right sides of the rectangle, respectively. Here is an example:



The leftmost disk should be centered at the origin. Assume that functions `DrawRect` and `DrawDisk` are available. Examples of their use are given below:

```
DrawRect(0,1,5,2,'y') % A yellow rectangle of length 5 and width 2
                    %   with its lower left corner at (0,1)
DrawDisk(3,4,6,'w')  % A white circle of radius 6 centered at (3,4)
```

```
r= W/2; % radius of circle

DrawRect(-r,-r,L,W,'y')

N = floor(L/W); % Number of disks

S = (L-W)/(N-1); % center to center spacing

for k = 1:N

    x = (k-1)*S;

    DrawDisk(x,0,r,'w')
end
```