

Name: _____
 (Legibly print last name, first name, middle name)

NetID: _____

Statement of integrity:

I did not, and will not, violate the rules of academic integrity on this exam.

_____ (Signature)

Q1: (25)	_____	_____
Q2: (25)	_____	_____
Q3: (25)	_____	_____
Q4: (25)	_____	_____
Total: (100)	_____	_____

Circle your lecture time: 9:05 or 11:15

Circle your section instructor's name:

	<i>Tuesday</i>	<i>Wednesday</i>
10:10		Chris Scheper
11:15		Lucian Leahu
12:20	Lucian Leahu	Chris Scheper
1:25	Lucian Leahu	Chris Scheper
2:30	Mateo Restrepo	Mateo Restrepo
3:35	Mateo Restrepo	Kenneth Tsung

Instructions:

- This is a 90-minute, closed-book exam; no calculators are allowed.
- The exam is worth a total of 100 points, so you should try to spend no more than about 18 minutes on a 20-point question.
- Read each problem completely, including any provided code, before starting it.
- Raise your hand if you have any questions.
- Use the backs of pages or ask for additional sheets of paper as necessary.
- Clarity, conciseness, and good programming style count for credit.
- If you supply multiple answers, we will grade only *one*.
- Use only MATLAB code. No credit for code written in other programming languages.
- Assume there will be no input errors.
- Write user-defined functions only if asked to do so.
- **Do not use arrays.**
- Do not use `switch`, `try`, `catch`, or `break` statements.
- You may find the following MATLAB predefined functions useful:
`abs`, `sqrt`, `rem`, `rand`, `floor`, `ceil`, `input`, `sprintf`, `disp`

Examples: `rem(5,2)` → 1, the remainder of 5 divided by 2
`rand` → a random real value in interval (0,1)
`floor(6.9)`, `floor(6)` → 6, rounds down to the nearest integer
`ceil(8.1)`, `ceil(9)` → 9, rounds up to the nearest integer

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

y

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

y

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
```

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)
```

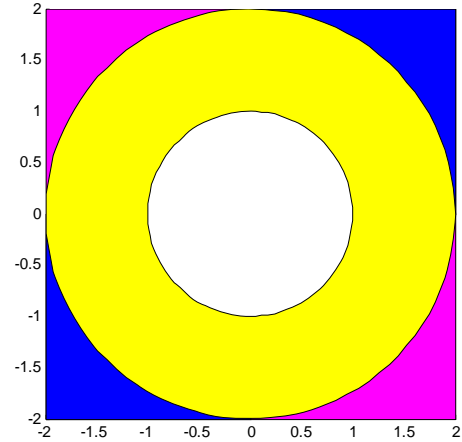
Part (e): (8 points)

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 _____  
    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 disks ...
% Code not shown. Assume it is correct.

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

x =

y =

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

Question 3: (25 points)

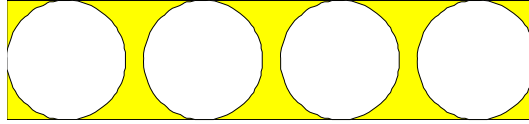
The value of $\frac{\pi^2}{8}$ may be approximated by the series

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

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.

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)
```
