CS 1112 Spring 2020 Prelim 1 March 10 7:30–9:00 PM

Name: ______________________________ NetID: __________

(Legibly print last name, first name, middle name)

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

____

(Signature)

Circle your discussion instructor's name:

10:10		Qinru Shi
11:15		Qinru Shi
12:20	Noam Eshed	Noam Eshed
1:25	Zhilong Li	Noam Eshed
2:30	Zhilong Li	Vaishnavi Dhulkhed

Instructions:

- Check that this packet has 8 double-sided sheets.
- This is a 90-minute, closed-book exam; no calculators are allowed.
- The exam is worth a total of 100 points, so it's about one point per minute!
- Read each problem completely, including any provided code, before starting it.
- Raise your hand if you have any questions.
- Use the back of the pages or ask for additional sheets of paper as necessary
- Clarity, conciseness, and good programming style count for credit.
- Indicate your final answer. If you supply multiple answers, you may receive a *zero* on that question.
- Use only MATLAB code. No credit for code written in other programming languages.
- Assume there will be no input errors.
- Write user-defined functions and subfunctions only if asked to do so.
- Do not use `switch`, `try`, `catch`, `break`, `continue`, or `return` statements.
- Do not use built-in functions that have not been discussed in the course.
- You may find the following MATLAB predefined functions useful: `abs`, `sqrt`, `log`, `rem`, `floor`, `ceil`, `rand`, `zeros`, `ones`, `linspace`, `length`, `input`, `fprintf`, `disp`

Examples:

`rem(5,2)` → 1, the remainder of 5 divided by 2
`rand()` → a random real value in the interval (0,1)
`sqrt(16)` → 4, square root
`abs(-3)` → 3, absolute value
`log(1)` → 0, natural logarithm
`floor(6.9)`, `floor(6)` → 6, rounds down to the nearest integer
`ceil(8.1)`, `ceil(9)` → 9, rounds up to the nearest integer
`length([2 4 8])` → 3, length of a vector
`zeros(1,4)` → 1 row 4 columns of zeros
`linspace(3,5,10)` → a vector of 10 real numbers evenly distributed in the interval [3,5]

Use this page for scratch work.

Question 1. (13 points)

(a) What will be printed when the following code is run? Pay attention to the alignment produced by the format specifier and clearly show which letters/digits are on top of which (if any), assuming all letters/digits/spaces have the same printed width.

```
m= 3;
n= 2;
rice= 1;
for k = 1:m*n
    fprintf('%3d', rice)
    if rem(k,m) == 0
        fprintf('\n')
    end
    rice= rice*2;
end
```

Output:

(b) What will be printed when the following script is executed?

Script	*Function* (in `foo.m`)
`v = [4 5 10];`	`function a = foo(b,a)`
`k = [2 3 1];`	
`a = v(k(2));`	`b = a - b;`
`fprintf(’a is %d\n’, a)`	`a = b^2;`
`b= 6;`	
`c= foo(a,b);`	`fprintf(’b is %d\n’, b)`
`fprintf(’c is %d\n’, c)`	`fprintf(’a is %d\n’, a)`
`fprintf(’a is %d\n’, a)`	

Output:

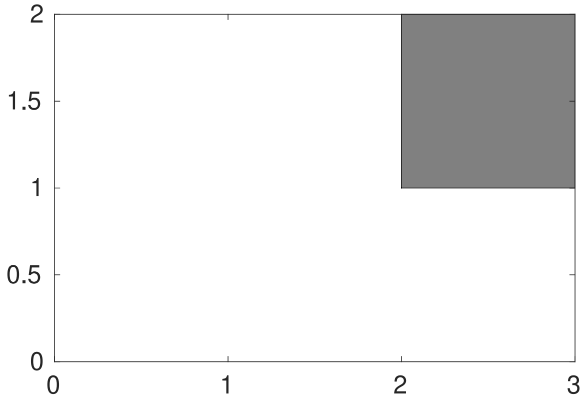
Use this page for scratch work.

Question 2. (17 points)

(a) Fill in the blank so that the code snippet keeps generating random points within the figure area until a point lies in the shaded region (whose lower-left corner is at $(2, 1)$). The boundary of the shaded region counts as "inside" the region.

```
x= 3*rand();
y= 2*rand();

while _________________________
    x= 3*rand();
    y= 2*rand();
end
```

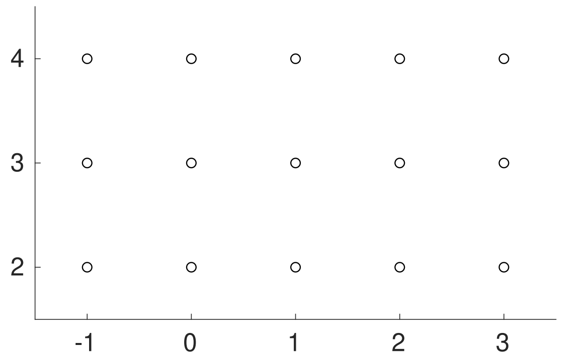


(b) Write a code snippet that assigns to variables x and y the coordinates of one point chosen randomly from the set of points shown in this plot (observe that the coordinates of the points are all integers):

```
% Complete these assignments


x=


y=
```



(c) Fill in the blanks in the range expression below so that the vector w will contain the same values as vector v (ignoring rounding error). Assume that a < b and that n is an integer greater than 1. You may define additional variables above the assignment to w if you wish.

```
v = linspace(a, b, n);



w = a : _____________________ : _____________________ ;
```

Use this page for scratch work.

Question 3. (24 points)

An automotive consultant has determined that car buyers care most about cost, speed, and safety, which can be computed as scores (unitless point values) in a simplified model from the engine horsepower h, car frame weight w in kilograms, and passenger capacity c as follows:

$$\text{speed} = \frac{\log(h)}{cw^2} \qquad \text{cost} = \sqrt{hc} \qquad \text{safety} = 10\frac{w}{c}$$

(here, $\log(x)$ refers to the natural logarithm, following MATLAB's convention).

(a) Implement the following function as specified:

```
function [speed, cost, safety] = calc_scores(c, w, h)
% Compute the speed, cost, and safefy scores from the passenger capacity
% `c`, car frame weight `w`, and engine horsepower `h` according to the
% consultant's model.
```

Use this page for scratch work.

(b) By making effective use of function calc_scores(), implement the following function as specified:

```
function [max_speed, best_h, best_w] = optimize_car(target_c, target_s)
% Iteratively find the combination of horsepower (best_h) and weight (best_w)
% that achieves the highest speed score (max_speed) in the design of a
% 4-passenger car with the the following constraints:
% - possible values of engine horsepower are 50, 51, ..., 200
% - possible values of car frame weight are 1500, 1600, ..., 3000
% - cost score of the design cannot exceed the target cost score (target_c)
%   by more than 20 points
% - safety score of the design cannot differ by more than 30 points from
%   the target safety score (target_s)
% If multiple combinations of horsepower and weight result in the highest
%   speed, any one of those combinations may be returned.
% If no combination of horsepower and weight can meet the constraints, then
%   set all the return parameters to 0.
```

Use this page for scratch work.

Question 4. (22 points)

(a) Consider a game in which a hero is fighting a monster. In each round of the game, the hero takes an action, and then, if the monster is still alive, the monster takes an action. The monster's action is always to attack the hero, dealing **20** points of damage, which is deducted from the hero's pool of hit points (initially **100**). The hero may either attack or heal for their action: healing will add **50** hit points back to their pool (which is allowed to exceed its initial value), while attacking will do **30** damage to the monster (who starts with **200** hit points). The hero and monster are alive as long as their remaining hit points are greater than zero.

Write a function named `simCombat` to simulate playing this game when the hero decides between their two actions *randomly.* Your function must accept a single input parameter—the probability that the hero chooses to attack (rather than heal)—and must return a vector containing the hero's health after each round. Specifically, assuming the result is stored in a variable `hps`, then `hps(k)` is the number of hit points in the hero's pool after `k` rounds have been played (hit points may be negative). Your function should simulate rounds of combat until either the hero or the monster has died.

Use this page for scratch work.

(b) Complete the if condition so that the following snippet prints the appropriate message based on whether or not the hero survives combat. You may define additional variables above the if-statement if you wish.

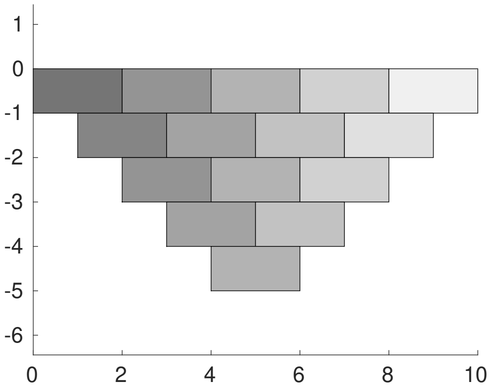
```
% Play a game of hero-fights-monster in which hero attacks
% with 95% probability.
hps= simCombat(0.95);

% Print the outcome of the game


if _______________________________________________
    % Hero survived
    disp('Our hero has vanquished the monster!')
else
    % Hero did not survive
    disp('Our hero has fallen to the monster.')
end
```

Use this page for scratch work.

Question 5. (24 points)



Complete a script to draw an upside-down pyramid composed of bricks. The script prompts the user for the number of bricks in the top row and for the width of a single brick. The upper-left corner of the top-left brick should be at position $(0, 0)$, and each brick should have a height of 1. Each lower row should have one fewer brick than the row above it, and its bricks should be centered beneath the edges of the bricks above (see figure).

The bricks should be colored brighter the further they are to the right. Specifically, their color should be *linearly interpolated* between black and white according to the x-coordinate of their midpoint: a brick whose midpoint is at $x = 0$ should be black, while one whose midpoint is at the right edge of the figure should be white (note: no bricks in the pyramid actually have their midpoints at these extremes).

To draw the bricks, assume the existence of the following function (do not implement it yourself):

```
function DrawRect(x0, y0, width, height, color)
% Draw a rectangle and fill it with color.
% The lower-left corner of the rectangle will be at (`x0`,`y0`), its
% width and height will be `width` and `height`, respectively, and it
% will be filled with the RGB color specified by the 3-vector `color`.
```

We've provided the graphics setup and input for you; write the rest of the script in the space below:

```
axis equal
hold on
black= [0 0 0];     white= [1 1 1];
n= input('Enter number of bricks in top row: ');
w= input('Enter width of a brick: ');
h= 1;  % Brick height
```

```
hold off
```

Use this page for scratch work.