**JUL 20 Data manipulation, nested loops**

**Concepts**

Control flow

Branching: if statements

Nested if statements

Looping: while/for loop

Control manipulation (break, continue)

Nested loops

For-while translation

Array

Size (dynamic, not fixed; higher dimensions), length, numel

Subarray assignment/creation

Induction

Base case

Induction hypothesis

Inductive step

**Q4:**

Write the following function:

function [rvec, cvec] = findInMatrix(n,M)

% Find all occurrences of the number n in matrix M.

% rvec and cvec are column vectors of row and column numbers such that

% M(rvec(k), cvec(k)) is equal to n.

% If n is not found in M, rvec and cvec are empty vectors.

Use loops in this problem; do NOT use the built-in function find.

**Q5:**

Write the following function:

function A = matrixCSums(M)

% M is a numeric matrix and A has the same size as M.

% Each element in A is the sum of the corresponding element in M

% and all the elements above it. Example:

% M = [ 1 3; ... A = [ 1 3; ...

% 4 5; ... then 5 8; ...

% -7 2]  -2 10; ...

Do NOT use any built-in functions other than size.

**Q6:**

What is the output from executing the following script? If the program doesn’t terminate or if there will be an error during execution, write the word “error” instead of the output.

v = [3 1 4 2];

for k = 1:4

v( v(k) ) = v(k);

end

disp(v)

**Q7:**

Write a boolean expression on the blank below so that the resulting fragment keeps prompting the user to enter a number until an appropriate number is entered or until 10 numbers have been entered, whichever occurs first.

n = input(’Enter a value that is either negative or greater than 500: ’);

k = 1;

while \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

n = input(’Enter a value that is either negative or greater than 500: ’);

k = k+1;

end

**Q8:**

Complete the fragment below to draw as many non-overlapping white disks of diameter W as possible within an L-by-W rectangle, 0 < W < L/2. 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. The leftmost disk is centered at (0,0). Assume the availability of the function DrawDisk. For example, DrawDisk(3,5,2,’w’) draws a white disk of radius 2 centered at (3,5). Do NOT use built-in function linspace.



figure;

axis equal off;

hold on;

L = input(’What is the length L of the rectangle? ’);

 W = input(’What is the width W of the rectangle (W<L/2)? ’);

 r = W/2; % radius of each disk

DrawRect(-r,-r,L,W,’y’)

% yellow L-by-W rectangle with lowerleft corner at (-r,-r)

n = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; % number of disks

g = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; % gap between two disks

for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

end

hold off