**JUL 13 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

**Q1:**

The following program reads an integer k and outputs all the positive multiples of k up to 1000. Fill in the blank.

k = input(’Please enter a positive integer smaller than 1000: ’);

for j = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fprintf(’%d ’, j);

end

fprintf(’\n’);

**Q2:**

Complete the following function so that it performs as specified:

function x = IsPythag(a,b,c)

% x has the value of 1 if a triangle with sides a, b, and c is

% a Pythagorean triangle and 0 otherwise.

% a, b, and c are positive integers.

**Q3:**

Write a function y = Mid3(a,b,c) that returns the middle of the three values a, b, and c.

**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.