**CS1112 Exercise 6** Name: _____ NetID: _____

You have until *Monday, 3/2, at 9pm* to complete this exercise and submit Problems 2, 3, and 4 using MATLAB *Grader*. Problem 1 does not need to be submitted or checked off (but you *do* need to think about what's going on).

# 1 Different ways to create vectors

Type the following expressions in the MATLAB *Command Window* to see what kind of vectors they create. Write the resulting vectors (and answer the questions) on the blanks.

```
a= zeros(1,4)  %_____________

b= zeros(4,1)  %_____________  What do the arguments specify?_____________________

c= ones(1,3)   %_____________

d= 10:2:17     %_____________

f= 10:-1:17    %_____________

g= linspace(10,19,4) %_____________

h= linspace(10,6,5)  %_____________

k= [10 20 40]  %_____________  What does the space separator do?_________________

m= [10,20,40]  %_____________  What does the comma separator do?_________________

n= [10;20;40]  %_____________  What does the semi-colon separator do?_____________

p= [a k]       %_____________

q= [b; n]      %_____________

r= [a n]       %ERROR--mismatched dimensions! (Attempt to concatenate a column to a row)

s= b'          %_____________  This operation is called "transpose"

t= [a b']      %_____________
```

# 2 Basic loop pattern for a vector

**Solve in MATLAB Grader**

(a) Accumulation Pattern: Compute the sum of all the elements in vector `v`. Do not use built-in function `sum()`.

(b) Finding the best in a set: Find the maximum value in vector `v`. Do not use built-in functions `max()`, `min()` and `sort()`.

# 3 Searching within a vector

**Submit solution in MATLAB Grader**

Write a function `vectorQuery(v,n,r)` to determine whether the number `r` appears in the first `n` components of vector `v`. (Assume that `r` is an integer and that `v` stores integer values.) The function returns `true` if `r` is in the first `n` components of `v` and `false` otherwise. Your function assumes that `v` is a vector of numbers, `n` is a positive integer, and `r` is a number. Use a loop to do the search. (Do not use `find()`, `contains()`, or vectorized code.) Make sure that the loop index doesn't go "out of bounds" (if `n` is greater than the length of vector `v`). *Be efficient*: the loop should stop as soon as `r` is found.

## 4 Creating vectors of unknown length

**Submit solution in MATLAB Grader**

Write a function `sequence(m)` that generates a sequence of random *integer* numbers between 1 and $m$, inclusive, stopping when a value is repeated for the first time. $m > 1$. The function returns a vector containing all the numbers generated (in the order in which they were generated) except for the last value that is a repeated occurrence.

Example: If the generated sequence is 3 1 9 5 7 2 5, the vector to be returned should be 3 1 9 5 7 2.

*Notes:* 1) Use built-in functions `rand()`, `floor()`, `ceil()` to generate random integer values; *do not* use function `randi()`. 2) Use a `while`-loop since this problem is a case of indefinite iteration—the number of iterations needed is not known in advance. 3) Make effective use of the function `vectorQuery()` that you have developed already—*Do not* use built-in functions `find()` or `contains()`. 4) When you don't know how long a vector needs to be, you can build it one component at a time. Here is an example to store only the even integer values that a user enters:

```
% Prompt user to enter positive integers and store the even integers in
% a vector v.
k= 0;  % vector length so far
num= input('Enter a positive integer: ');
while num > 0
    if rem(num,2) == 0
        k= k + 1;
        v(k)= num;
    end
    num= input('Enter a positive integer (negative to stop): ');
end
```

**Be sure to log off the lab computer before you leave the lab.**