CS1112 Lab Exercise 6

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

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

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

m= [a g]       %_____________

n= [b; k]      %_____________

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

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

r= [a b']      %_____________
```

2 Basic loop pattern for a vector

(a) Given a vector `v`, display the values stored in the vector one at a time (one number on each line).

(b) Given a vector `v`, display the maximum value stored in the vector. Do not use built-in functions `max` and `min`.

3 Examining a subarray

Write a function `vectorQuery(v,n,r)` to determine whether the number `r` appears in the first `n` components of vector `v`. The function returns 1 if `r` is in the first `n` components of `v` and 0 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` 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`).

4 Creating arrays of unknown length

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. The function returns an array containing all the numbers generated (in the order in which they were generated) except for the last value that is a repeated occurrence.

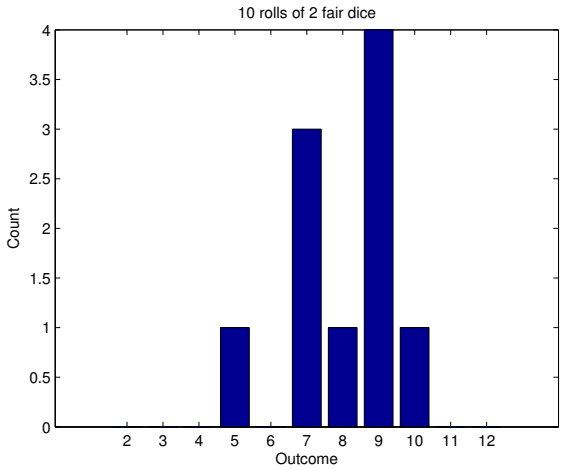
Example: If the generated sequence is 3 1 9 5 7 2 5, the array to be returned should be 3 1 9 5 7 2.

Hints: 1) Use the function `vectorQuery` that you have developed already. 2) 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 numbers and store the even integers in a vector v
k= 0;  % vector length so far
num= input('Enter a positive number: ');
while  num>0
    if  rem(num,2)==0
        k= k+1;
        v(k)= num;
    end
    num= input('Enter a positive number (negative to stop): ');
end
```

5 Roll multiple dice

Start by reviewing the function `rollDie` (Lecture 10) which simulates the rolling of one fair, six-sided die. Next, write a function `rollDice(n,d)` to simulate the rolling of `d` six-sided dice `n` times. We define the *outcome* of rolling `d` dice once to be the sum of the faces that show up. In the function, create a vector `count` such that `count(c)` is the number of times that outcome `c` has occurred. *Do not* use built-in function `sum`. Your function draws a histogram of the result. Below is an example histogram for small `n`. What shape do you expect to see for large `n`?



Please delete your files from the computer before you leave the lab.