CS1112 Exercise 7 Name: _____ NetID: _____

You have until *Sunday, 3/8, at 9pm* to complete this exercise and get it checked off (during this discussion section or during *consulting hours* or *TAs' office hours*).

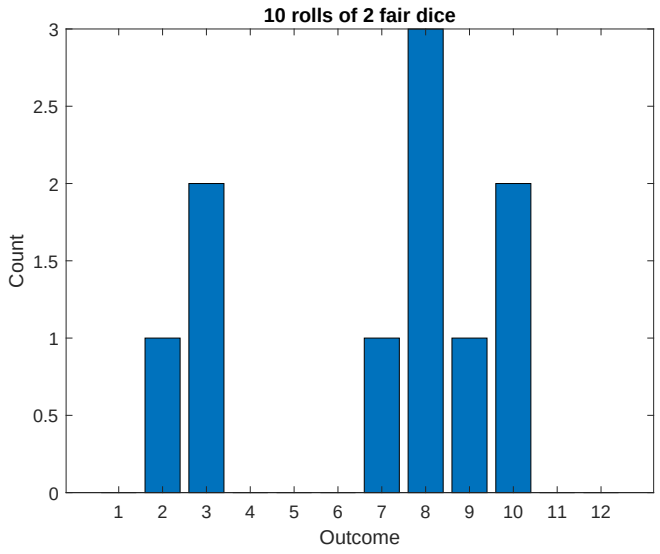
1 Different ways to create vectors

For each statement below, 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)   %_________________

f= 10:-1:17    %_________________

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

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

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

p= [a k]       %_________________

q= [b; n]      %_________________

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

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

2 Roll multiple dice

Review the function `rollDie` (from Lecture 11; see back), which simulates the rolling of one fair six-sided die. Then write a function `rollDice(n,d)` to simulate the rolling of `d` six-sided dice `n` times and draw the resulting histogram. We define the *outcome* of rolling `d` dice once to be the sum of the faces that show up. The function returns the vector `count`, where `count(c)` is the number of times that outcome `c` has occurred. For extra practice with the accumulation pattern, *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`?



```
function count = rollDie(rolls)
% Simulate rolling a fair 6-sided die and draw histogram of outcomes
%   `rolls` is the number of times to roll the die
%   `count` is a vector of how many times each outcome occurs
%     `count(f)` is the number of times face `f` occurs

FACES= 6;               % number of faces on die
count= zeros(1,FACES);  % bins to store counts

% Count outcomes of rolling a FAIR die
for k = 1:rolls
    % roll the die
    face= ceil(rand()*FACES);
    % increment appropriate bin
    count(face)= count(face) + 1;
end

% Show histogram of outcome
bar(1:FACES, count)
title(sprintf('Outcomes from %d rolls of a fair die', rolls), 'Fontsize',14)
xlabel('Outcome', 'Fontsize',14)
ylabel('Count', 'Fontsize',14)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function count = rollDice(n,d)
% Simulate rolling `d` dice `n` times (trials) and draw histogram of outcomes
% Rolls all `d` dice in each trial; the outcome is the sum of their faces.
% `count` is a vector of the number of times each outcome occurs, i.e.,
%   `count(t)` is the number of times outcome `t` occurs

FACES= 6;                % six-sided dice
maxOut= FACES*d;         % highest possible outcome from rolling all dice
count= zeros(1,maxOut);  % bins to store counts
  % `count(c)` is the number of occurrences of outcome `c`
```