Homework 5  
Due October 1 in class  
20 points

Note: Please use good programming style. Comment your code. Use meaningful variable names. Use variables whenever a value is not simply a fixed constant in an equation. Separate logical sections of your code using %%.

Q1. Write code that implements randperm(). For now, you do not need to worry about implementing this in a function. A script will be fine.

Recall from class that if you type help randperm that
P = randperm(N) returns a vector containing a random permutation of the integers 1:N. For example, randperm(6) might be [2 4 5 6 1 3].

This section of your script should have a variable N that is set to some number and it should create a one-dimensional array that is a random permutation of the numbers 1:N.

I want you to implement two algorithms for producing a random permutation. These two algorithms are described on this Wikipedia page:
DO NOT search elsewhere on the internet for code that implements randperm(). This Wikipedia page is all that you are permitted to look at.

a. The first algorithm is called the Fisher–Yates Shuffle. As described on the Wikipedia page, the algorithm consists of the following steps:

1. Write down the numbers from 1 through N.
2. Pick a random number k between one and the number of unstruck numbers remaining (inclusive).
3. Counting from the low end, strike out the kth number not yet struck out, and write it down at the end of a separate list.
4. Repeat from step 2 until all the numbers have been struck out.
5. The sequence of numbers written down in step 3 is now a random permutation of the original numbers.

See the Wikipedia page for illustrations of the method. Try the algorithm by hand with a small value of N before trying to implement the algorithm in Matlab.

b. The second algorithm is Durstenfeld's method. The Wikipedia page show pseudocode for implementing this algorithm. Note that this pseudocode (like many examples of pseudocode you might find in books and online) assumed that arrays are indexed from 0 to N-1 (which is true in C/C++ and Python). I have rewrite the pseudocode here assuming that arrays are indexed from 1 to N (which is true for Matlab):
-- To shuffle an array a of n elements (indices 1..N):
for i from 1 to N-1 do
  j ← random integer such that i ≤ j ≤ N
  exchange a[i] and a[j]

As with the Fisher–Yates Shuffle, see the Wikipedia page for illustrations of Durstenfeld's method. Try the algorithm by hand with a small value of N before trying to implement the algorithm in Matlab.

**Q2.** Now imagine you have multiple blocks in an experiment, that within each block you have a randomization of nlevels of an independent variable, but that you need to make sure that the same level never repeats. In other words, if you had nblocks = 3 and nlevels = 3, then the following sequence of levels would be valid: 3 1 2 1 2 3 2 1 3. The following sequence of levels would be invalid because there are repeats: 1 3 2 2 1 3 3 1 2. Write Matlab code that creates this randomization, again using only rand() or randi() and control flow operators.

**Q3.** Imagine that a member of your lab was trying to re-create a bit of code that would generate random integers that are uniformly distributed in the closed interval [a,b] (meaning that the smallest number is a and the largest number is b) and then plot a histogram of them using the bar() plotting function. Here is their code:

```matlab
N = 100;
a = 2;
b = 6;
cnt = zeros(1, b);
for i=1:N
    rnd = a + round((b-a)*rand());
    cnt(rnd) = cnt(rnd) + 1;
end
bar(a:b, cnt(a:b));
```

**a.** Does this code produce random numbers uniformly distributed in the closed interval [a,b]? If yes, how do you know? If not, why not and fix the code.

**b.** Try the above code using a = -8 and b = -3. Fix the code so it works.

For both a and b, you cannot use irand(). You must use rand(). Also, you cannot use the histogram() or hist() functions or any other built-in functions.

*Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.*