MB208: Theoretical and computational neuroscience

Assignment 0: Basic programming with NEURON

Submission date: 22 Feb 2019

These are some simple programming problems, having nothing to do with neuroscience (except for using your own brain ☺), to get you started with NEURON as a programming environment. Look at online programmers' reference (http://www.neuron.yale.edu/neuron/static/docs/help/index.html) or the NEURON book for syntax and other details. Most of these should be solvable from examples given on the online programmers' reference, under specific functions that I have mentioned under hints with respect to each question! For this assignment, send me the codes for each of these questions put together as a ZIP/TAR.GZ (TGZ)/RAR file.

1. Create a file named "x.txt" and add sixteen numbers to the file with space or lines in between. In a HOC program, open this file and read these numbers into an array, and print these numbers onto standard output with spaces in between. Find the average and standard deviation of the 16 numbers and print them on to standard output (**Hint**: use for loops and the formulae for mean and SD).

2. Do the same thing as above using the Vector class, by reading these 16 numbers into a Vector. (**Hint**: You have specific functions in the Vector class for mean and SD. You don't have to write the code for computing them!)).

3. Pascal's formula says:

$$ {}_nC_r = {}_{(n-1)}C_r + {}_{(n-1)}C_{(r-1)} \qquad (1) $$

   Pick an arbitrary value for *n:* $n \geq 10$. Prove empirically that equation (1) holds for all *r* within the range $1 \leq r \leq n$ (use a for loop for testing for different *r*'s in that range), using the definition of the binomial coefficient:

$$ {}_nC_r = \frac{n!}{r!\,(n-r)!} \qquad (2) $$

   Do this by writing a function for ${}_nC_r$ (with two arguments *n* and *r*), and calling it once for obtaining the LHS, and twice for obtaining the RHS of equation (1). Print an error if LHS is not equal to RHS for any of the *r* values. If LHS equals RHS **<u>for all $1 \leq r \leq n$</u>**, then print a statement saying that the Pascal's formula held true for all $1 \leq r \leq n$, for the chosen value of *n*.

   Note that a call of the function for ${}_nC_r$ can in turn call another factorial function thrice for computing equation (2). Make sure that you take care that 0! = 1, ${}_0C_r = 0$, ${}_nC_0 = 1$, ${}_nC_r = 0$ if $n < r$ and ${}_nC_n = 1$.

4. Using *double* arrays, and sort the 16 numbers you read in 1 in descending order and print them to standard output. (Look for sorting algorithms in the net; use any sorting algorithm of your choice.).

5. Do the same as 4 using the Vector class. (**Hint**: Use the sort function in the Vector class).

6. Create a file and type in a set of strings (names of your class mates, say). Read them into the program, and sort them in alphabetical order, and print to standard output the sorted and unsorted versions (**Hint**: use ropen() and getstr() for reading, and strcmp() for comparing strings; the rest of the sorting procedure would be the same as sorting numbers. **Hint to Hint:** You will see that it is not that straightforward ☺. You will need to come up with your own ingenious solution!).