**[Dan Stribling](https://app.slack.com/team/U010LBAS64Q" \t "_blank)**  [1:35 PM](https://covidcompbio.slack.com/archives/C01181TL7ME/p1586799355003400)

"Homework" from office hours:

A. Given the following nucleotide sequence:  
"ACTCCCGGATCTGCGGATCG"  
Create a function that calculates the nucleotide frequency of each nucleotide...

1. (Easy) Using the .count() and len() functions
2. (Int.) Using a for loop to iterate over every nucleotide by index number, and adding to the nucleotide count at that index. (Hint: use range(len(seq)) with the for loop)

B. Create a list with the following nucleotide sequences:

* ATGCGGCTTAT
* GCTAAATC
* GTTTTTTTCTT

1. (Easy) Directly, using [seq1, seq2...]
2. (Int.) Starting with an empty list, and using the .append() method
3. (Hard) Starting with a list of the above sequences in uppercase, create a list of the sequences in lowercase using the .lower() function with a list comprehension. (will require googling)

C. Using a list of all sequences from part A and B, perform the following actions:

* Print the sequence and sequence length on one line.
* If the sequence contains the subsequence "GCTT", print "Contains Subsequence [subsequence]" via...:

1. (Easy): using "in"
2. (Hard): By checking each length 4 substring in the string using string slicing (would require googling)

* If the sequence ends with the subsequence "TCTT", print "Ends With Subsequence [subsequence]" via...:

1. (Easy):  the .endswith() method
2. (Hard): Using len() and string slicing (requires googling)

D. (ADVANCED): Create a function that performs the same task as part C. Instead of using a specific list of sequences, and a specific subsequence, code the function to take any list of sequences and any subsequence for its testing.

1. (Very Hard) Run the function with the data from part C
2. (Very Hard) Create a new dataset, and run the function
3. (Very Hard) Verify the results are as you expect.

(edited)