

1. Hints for HW #1:
 - use `ord()` function to convert character to its numeric ASCII value
 - scale the ASCII values for the upper-case letters to values in the range 0_{10} (e.g., 'A' $\rightarrow 65_{10} \rightarrow 0_{10}$) to 25_{10} (e.g., 'Z' $\rightarrow 90_{10} \rightarrow 25_{10}$) by subtracting by 65_{10}
 - this allows us to add the shift amount and wrap back around in the alphabet by using the remainder operator `%` (e.g., Assume the shift amount is 3 for 'Z': 'Z' $\rightarrow 90_{10} \rightarrow 25_{10} \rightarrow (25_{10}+3) \% 26 \rightarrow 2_{10}$)
 - scale back to the ASCII values by adding 65_{10}
(e.g., Assume the shift amount is 3: 'Z' $\rightarrow 90_{10} \rightarrow 25_{10} \rightarrow (25_{10}+3) \% 26 \rightarrow 2_{10} \rightarrow 67_{10}$)
 - use the `chr()` function to convert the numeric ASCII value to the corresponding encoded character
(e.g., Assume the shift amount is 3: 'Z' $\rightarrow 90_{10} \rightarrow 25_{10} \rightarrow (25_{10}+3) \% 26 \rightarrow 2_{10} \rightarrow 67_{10} \rightarrow$ 'C')
2. `input` function for numeric values vs. `raw_input` function for strings
3. Motivation of looping in an algorithm (e.g., walking example)
4. for-loop iterating over a Python list or string
5. `range` function to generate an integer list
6. `xrange` function to generate a sequence of integer values for the for-loop without actually creating a list
7. Example: counted-controlled for-loop to calculate the average
8. Augmented assignment operations (e.g., `myIntVariable += 1`)