Time complexity and List exercises (do these in any order you wish)

Note: If you aren't quite sure what the time complexity of the operation is feel free to jump on discord and we can take a better look at pinpointing the correct time complexity

- **Power (also done in teacher's video)**
  - Make a function that takes two parameters (the second one a positive integer)
  - Return the result of raising the first parameter to the power of the second
  - Use only math operators, not built in Python functions
  - *What is the time complexity of your solution for this operation?*
    - *Ísl: hver er tímaflækja ykkar lausnar á þessari aðgerð?*
    - *What is 'n' in regard to the time complexity?*
  - ***ADVANCED:** Can this function be done at a better time complexity?*
    - *If so what is it?*
    - *Describe a method that would do it at a better time complexity*
    - *It is not necessary to implement this, and it is recommended to **save this until last**, if you still have time after the other parts of this assignment.*

- **Multiplication of positive integers**
  - Make a function that takes two parameters
  - Return the result from multiplying the two integer parameters with each other
  - The only math operators you may use in your implementation are + and -
  - *What is the time complexity of your solution for this operation?*
    - *Ísl: hver er tímaflækja ykkar lausnar á þessari aðgerð?*
    - *What is 'n' in regard to the time complexity?*

- **Random number insertion**
  - Build a list of a certain size (lis = [0] * size)
  - Traverse list and for each location in the list:
    - Random generate number between 1 and 6
    - Put the number in the "current" location
  - *What is the time complexity of this operation?*
    - *Ísl: hver er tímaflækja þessarar aðgerðar?*
    - *What is 'n' in regard to the time complexity?*

- **Print your list to the screen**
  - Traverse the list and for each element in the list
    - Print the number to the screen
    - Make the whole list display in one line
    - Separate with both comma and space
      - 3, 6, 1, 8, 3
    - Make sure there is not a comma at the end

    - Make a function that does this for you
        - The function can take a list as a parameter
    - *What is the time complexity of this operation?*
        - *Ísl: hver er tímaflækja þessarar aðgerðar?*
    - Now you can use this function to test all your other list operations, by outputting the results.

- **Increase numbers at random index**
    - Generate a random number from 0 up to the size of the list
    - At the element at that location in the list
        - Increase the number at that location by 1
    - *What is the time complexity of this operation?*
        - *Ísl: hver er tímaflækja þessarar aðgerðar?*
    - Run a program that tests this several times
        - *So many times that you're sure the random location picks every location in the list at least once (it's OK if the list used is short)*

- **Switch items in list**
    - Generate and fill a list of some size
    - Pick a location in the list
        - Switch the value in that location with the location next to it
    - Randomly pick two locations in the list
        - Switch the values in the two locations
    - *What is the time complexity of this operation?*
        - *Ísl: hver er tímaflækja þessarar aðgerðar?*

- **Ordered insertion**
    - Random generate a number between 1 and 6
    - Put the number in the correct location so that the list is ordered
- *What is the time complexity of your implementation of this operation?*
    - *Ísl: hver er tímaflækja ykkar útfærslu á þessari aðgerð?*
    - *Answer for both a single insertion and for populating of the whole list with n items*

- **Combined insertion and ordering**
    - First fill a list with random number insertion
    - Then for each element in that list
        - Add the number into another list with ordered insertion
    - *What is the time complexity of this operation?*
        - *Ísl: hver er tímaflækja þessarar aðgerðar?*
    - Now try to implement this operation without ordering into a different list, instead working only with the items within the list.
        - **Hint:** imagine that each item is being inserted into the list to the left of it, starting with the second element, then choosing the next element to the right and inserting it into the sorted list to the left of it.