Runtime Analysis

	doublerAppend - push	doublerInsert - unshift
tinyArray	154.479 μs	66.112 μs
smallArray	217.175 μs	97.511 μs
mediumArray	281.993 μs	316.177 μs
largeArray	1.294062 ms	21.514483 ms
extraLargeArray	6.061099 ms	2.058856277 s

HOW DOES EACH FUNCTION “SCALE”?

doubleAppend scales linearly because it only relies on the length of the array passed to it

doubleInsert scales quadratically because it uses unshift, this means that for every element in the array passed as argument it will re-arrange all the elements in the **new_nums** array essentially doubling the workload.

WHICH OF THE TWO FUNCTIONS SCALES BETTER? HOW CAN YOU TELL?

doubleAppend scales better because it has a time complexity of **o(n)** while **doubleInsert** has a time complexity of **o(n^2)**