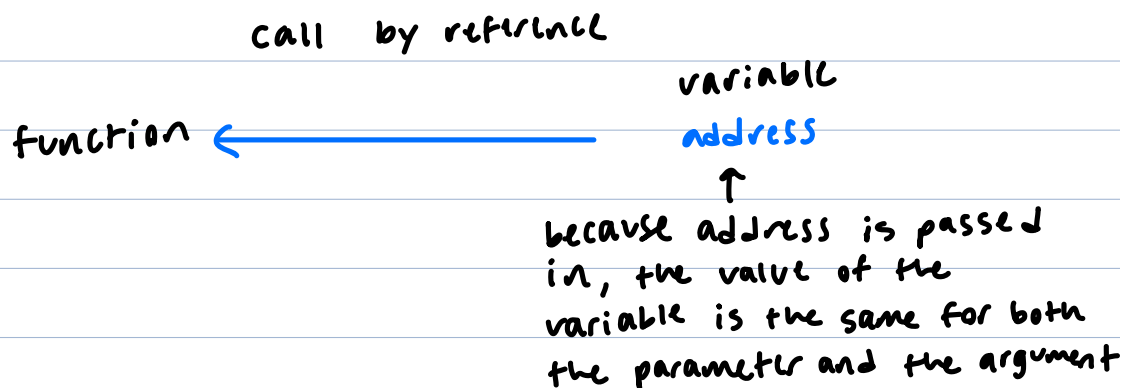
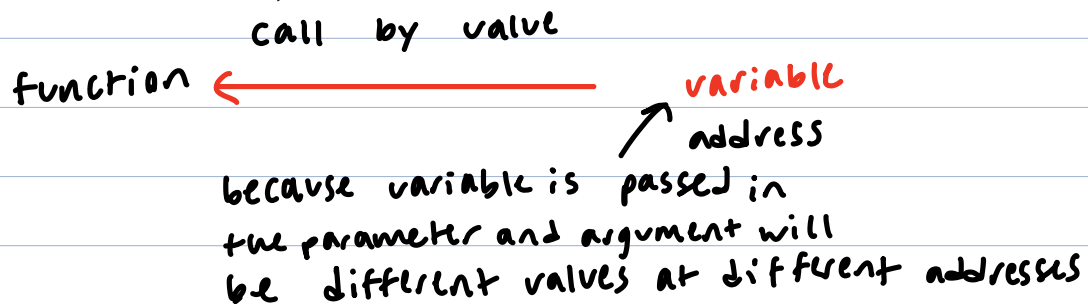


1. Call by value mechanism will make changes on a copy of the argument provided in the function meaning that changes to a parameter of function main will not affect this argument. When call by value is used, parameters are stored in entirely different location (different address) in memory because they were copied. This causes changes made inside functions to not take place in the main function. On the other hand, the call by reference mechanism copies the address of an argument into a parameter which means that the changes made in individual functions will effect parameters in the main function. When call by reference is used, the exact same address is used for the parameters and the address which means when one value is modified, the other will be modified as well.



2. 0 iterations: { 5, 16, 2, 4, 14, 11 }  
1 iteration : { 2, 16, 5, 4, 14, 11 }  
2 iterations: { 2, 4, 5, 16, 14, 11 }  
3 iterations: { 2, 4, 5, 11, 14, 16 }

3. int theArray[5]

address	address	address	address	address
3124	3128	3132	3136	3140
↓	↓	↓	↓	↓
theArray[0]	theArray[1]	theArray[2]	theArray[3]	theArray[4]

because each index is 4 bytes (int = 4 bytes), the address increments by 4 after each index

memory address of theArray[2] is 3132