Kyle Shin
163007179
Programming Assignment 4: Simulating Caches

To simulate caches in this assignment, I implemented an array of linked lists that keeps track of the tag (and index + whole binary of the memory address in general but that's only because I free them whenever the node gets freed). Each individual node represents a block, linked list represents a set, and the array of linked lists representing a cache.

I know I could have done this more efficiently if I implemented a 2 dimensional array of the same nodes but it was easier to imagine the assignment with linked lists when thinking of what to do.

superFree - frees everything in the cache. O(XY) where X is the number of sets and Y is the size of the sets

numberOfTags - checks how many blocks are in a set, it is practically a helper function for inserting tag nodes. O(Y)

insertTagNode - places a tag node in fifo ordering while freeing the first tag node and all the elements of the node. O(Y)

checkTag - used to see if a hit or miss took place by going through the blocks of a given set. O(Y)