/*
 * Authors: Olaolu Emmanuel && Jordan Lyon
 */

SLCreate
space: size of two function pointers and a struct pointer. O(1)
efficiency: initialize all variables in struct. O(1)

SLDestroy
space: size of one two struct pointers. O(1)
efficiency: frees each node and then the struct. O(n) where n is the number of nodes.

SLInsert
space: size of one node. O(1)
efficiency: worst case is end of list insertion which is O(n)

SLRemove
space: size of two struct pointers. O(1)
efficiency: worst case is not finding the target which is O(n)

SLCreateIterator
space: size of one Iterator struct. O(1)
efficiency: O(1)

SLDestroyIterator
space: only space for a few pointers. O(1)
efficiency: iterator is of constant size O(1)

SLNextItem
space: only requires pointer to the data item O(1)
efficiency: uses the previous item's next-node value to find the next node. O(1)

SLGetItem
space: returns pointer. O(1)
efficiency: pointer is already in the struct O(1)