Problem 6: Union and Intersection
Explanation

Design Decisions:
Union function adds all values from llist_1 and llist_2 to the set, which doesn't allow for duplicates. Then transfers the values to a linked list.
Intersection function creates two sets and adds values from llist_1 to first set and values from llist_2 to second set. Then it uses set.intersection and transfers the values to a linked list.
I used sets because they don't allow for collisions.

Time Complexity:
$O(n)$ because in both functions, I only iterate over a single list at a time, and set functions use hash tables, so they're $O(n)$ in the worst case of collisions.

Space Complexity:
$O(n)$ because I use a new list to store the union values and intersection values (union_list and intersection_list)