NAME: JOSHUA ABUTO ID:1001530342

Task 1: test cases for your code

Function	Test case	Data/code	Does my code handle it? Here: handle= does NOT crash
sublist(link A, link pos_list)	Index out of bounds	A: 10 ->10 ->40 ->20 pos_list: (**-7**) -> 3 or pos_list: 3 -> **80000** -> 3 result: fct returns NULL	No
	A is NULL	link A = NULL; result: fct returns NULL	Yes
	A is empty	link A = new_list(); result: fct returns NULL	Yes
	pos_list is empty	link pos_list = NULL; result: fct returns NULL	Yes
	pos_list is NULL	link pos_list = newList(); result: fct returns NULL	Yes
	A is not modified by sublist(…) ….	A: 15 -> 100 -> 7 -> 5 -> 100 pos_list: 3 -> 0 ->2 result: A will still be : 15 -> 100 -> 7 -> 5 -> 100	Yes
	Normal data (as in hw writeup)	A: 15 -> 100 -> 7 -> 5 -> 100 -> 7 -> 30 pos_list: 3 -> 0 -> 6 -> 4	Yes
	Repeated position	A: 5 pos_list: 0 -> 0 -> 0 result: returns: 5-> 5-> 5	Yes
delete_occurrences (link A, int V)	Normal data, V is in A (as in hw write-up)	A: 15 -> 100 -> 7 -> 5 -> 100 -> 7 -> 30 V is 7, Result: A will become: 15-> 100-> 5 -> 100 -> 30	No
	V does not occur in A	A: 15 -> 100 -> 7 -> 5 V is 9, Result: A does not change: 15-> 100-> 7-> 5	Yes
	Repeated consecutive occurrences	A: 15 -> 7 -> 7 -> 5 V is 7, Result: A becomes: 15 -> 5	Yes
	A has one item and that is V	A: 7 V is 7 Result: A becomes Empty	Yes
	A has only items with value V in it	A: 7->7-> 7 V is 7 Result: A becomes empty	Yes
	A is NULL	A = NULL Result: A is not changed	Yes
	A is empty	A = new_list() Result: A is not changed	Yes

swap_first_third(link A)	A is NULL	A = NULL Result: A is not changed	Yes
	A is Empty	Result: A not changed	Yes
	A has the same number	A(0) = 7 A(2) = 7 Result: no change notice but swap occurs	Yes
	A has a length of two	Result: swap first and last position	No

CODE & DRAWING for swap_first_third (list A) (This is a reminder of what needs to be done. Do not squeeze the answer in here. Use an additional page.)

Task 2 :

Let N = |A| (N is the size of list A)

swap_first_third(link A): T(N) = N + 4 + 1

~~insertion_sort (link A) T(_____) = _____~~ Updated 2/12/19

delete_occurrences(link A, int V) T(N,V) = 3+N*(1+2)+1

sublist(link A, link pos_list) T(N,P) = 5+P*(1+P*(5)+5)+4

Task 3 (10 points) Given:

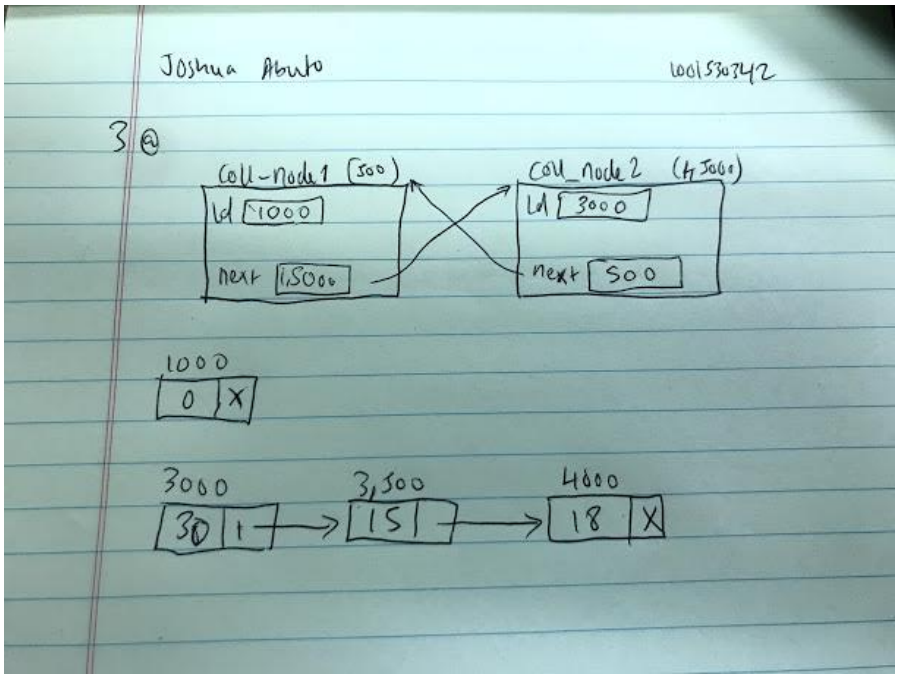
A new node structure (intended to be used to create a list of lists) is defined in the table below (using `struct node`):

```
struct node {
   int item;
   struct node * next;
};
```

```
struct coll_node  {
   link Ld; // NOTE: Ld must be represented with a dummy starting
   node. struct coll_node * next;
};
```

In your drawings, **show all the data as done in class** (including the list nodes, of type `struct node`). Use boxes for all member variables and write their value inside the box and their name outside the box.

a) (7 points) Draw two nodes (of type `struct coll_node`**)** that point to each other. For one of them Ld should be empty (but not NULL) and for the other one Ld should be: 30->15->18 . Use the representation with a DUMMY node for any normal list, Ld, part of nodes of type `struct coll_node`.



b) (3 points) Assume that an `int` is stored in 4 Bytes and a memory address is 8 Bytes. How much space will the above two nodes (and the data that they reference) occupy? That is, give the total space needed to store in memory what you drew above. **SHOW YOUR WORK**.