Q4

a) in1 = 1, 2, 3, 4 & in2 = 5, 6

#1 points to 1

in1->next =

llrec(in2, in1->next)

in2 points to 5; in1->next points to 2

#2 points to 5

in1->next =

llrec(in2, in1->next)

in2 points to 2; in1->next points to 6

#3 points to 2

in1->next =

llrec(in2, in1->next)

in2 points to 6; in1->next points to 3

#4 points to 6

in1->next =

llrec(in2, in1->next)

in2 points to 3; in1->next points to NULL

#5

in1 = 3

in2 = NULL

if(in2 == nullptr)

return in1;

Iteration	in	in1->next
#1	in1 = 1	5
#2	in1 = 5	2
#3	in1 = 2	6
#4	in1 = 6	3
#5	in1 = 3	4

bc the node that points to 3 (in1) still has the *next pointer set to 4

Final list returned:

1 -> 5 -> 2 -> 6 -> 3 -> 4 -> NULL

// each box represents an iteration of the llrec() function

// each box keeps track of the pointers that are passed on to the next iteration in the llrec() function that is returned

b) in1 = null
in2 = 2

```
llrec(in1, in2) {
    if (in1 == null) return in2;
    :
}
```

① in1 triggers the first if statement (base case)

② this if statement causes in2 to be returned

final list returned:
2 → NULL