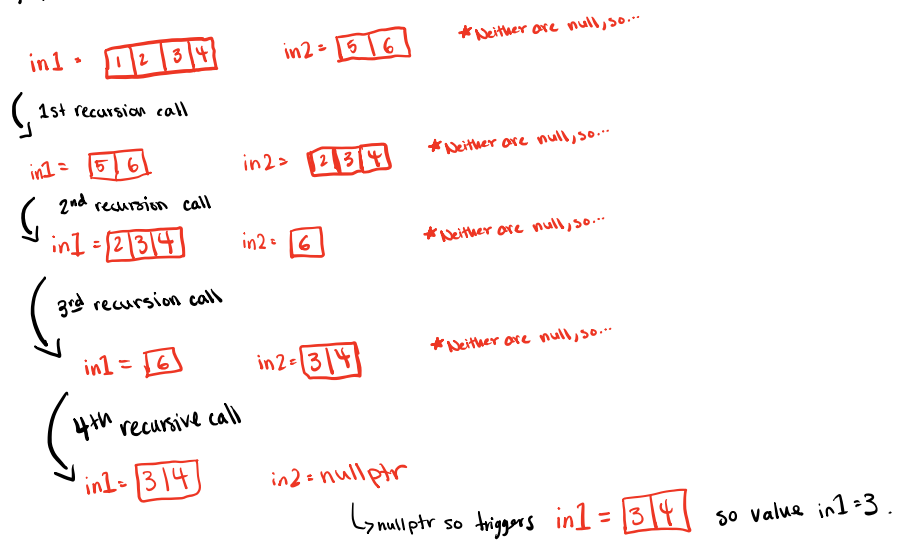```
struct Node {
    int val;
    Node* next;
};

Node* llrec(Node* in1, Node* in2)
{
    if(in1 == nullptr) {
        return in2;
    }
    else if(in2 == nullptr) {
        return in1;
    }
    else {
        in1->next = llrec(in2, in1->next);
        return in1;
    }
}
```

Question A: What linked list is returned if llrec is called with in1 = 1,2,3,4
and in2 = 5,6?

in1 = |1|2|3|4|    in2 = |5|6|    *Neither are null, so...

⤷ 1st recursion call

in1 = |5|6|    in2 = |2|3|4|    *Neither are null, so...

⤷ 2nd recursion call

in1 = |2|3|4|    in2 = |6|    *Neither are null, so...

⤷ 3rd recursion call

in1 = |6|    in2 = |3|4|    *Neither are null, so...

⤷ 4th recursive call

in1 = |3|4|    in2 = nullptr

⤷ nullptr so triggers  in1 = |3|4|   so value in1 = 3.

Question B: What linked list is returned if llrec is called with in1 = nullptr
and in2 = |2| ?

in1 = nullptr    in2 = |2|    in1 is equal to nullptr
                              which triggers a base case to
                              return in2.

⤷ Triggers Base Case.
  Returns in2 = |2| linked list