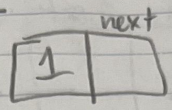
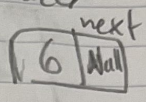
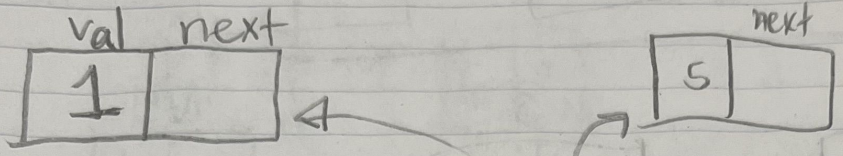
Problem 3 HW2 Ioannis Tsotras

a) n1 = &Node1 etc. n6 = &Node6

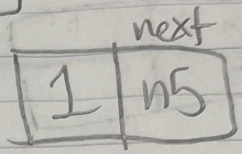




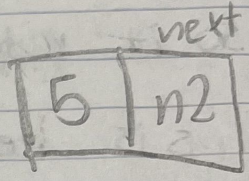


in1 = n1;

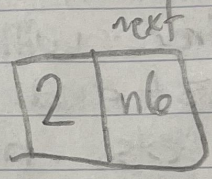
n1->next = llrec(n5, n2);
return n1;



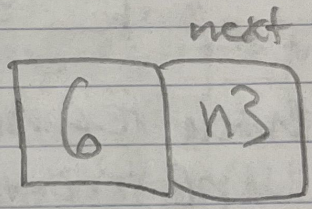
n5->next = llrec(n2, n6);
return n5;



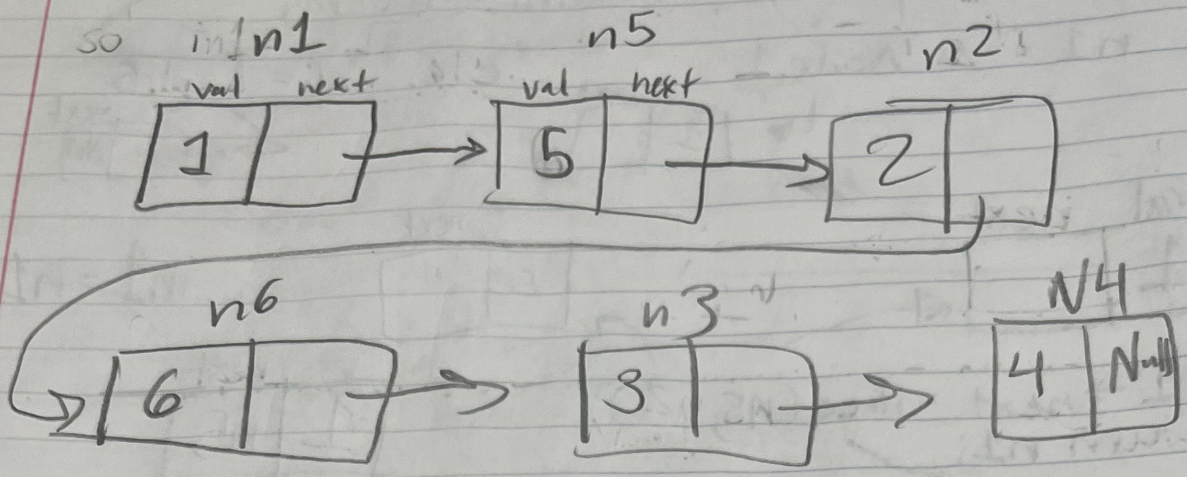
n2->next = llrec(n6, n3);
return n2;



n6->next = llrec(n3, Null);
return n6;



return n3; in2 == Null



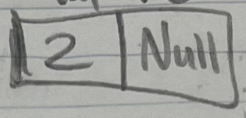
so it returns 1, 5, 2, 6, 3, 4

This is because the recursion recurses down each list to set its next pointer by calling Merge with the next node of the alternate list and the in1->next pointer. This eventually sets n5 as the second node and n6 as the 4th, while reordering n2 and n3 to be 3rd and 5th nodes, respectively.

When recursion reaches the base case, it sets
n6->next = n3; n2->next = n6;
n5->next->n2; and n1-> next=n5;

b) if(in1 == nullptr){
    return in2;
}

it returns the linked list



b/c in1 == nullptr