***Week – 3 (14.04.2021 – 18.04.2021)***

***CODES IN PDF***

1. ***Remove Nth Node From End of List:***

class Solution {

public:

ListNode\* removeNthFromEnd(ListNode\* head, int n) {

ListNode \*l = head;

ListNode \*l2 = head;

int i=0;

while(l!=NULL)

{

i++;

l=l->next;

}

if(i == n)

{

head = l2->next;

return head;

}

int j = i;

while(l2!=NULL)

{

j--;

if(j == n) l2->next = l2->next->next;

else l2 = l2->next;

}

return head;

}

};

1. ***Reverse Linked List:***

class Solution {

public:

ListNode\* reverseList(ListNode\* head) {

ListNode \*temp = NULL;

ListNode \*prev = NULL;

ListNode \*curr = head;

while(curr!=NULL)

{

temp = curr->next;

curr->next = prev;

prev = curr;

curr = temp;

}

return prev;

}

};

1. ***Reverse Linked List using Recursive Function:***

ListNode\* reverseList(ListNode\* head) {

if(head == NULL || head->next == NULL) return head;

ListNode \*temp = reverseList(head->next);

head->next->next = head;

head->next = NULL;

return temp;

}

};

1. ***Add Two Numbers II:***

class Solution {

public:

ListNode\* reverse(ListNode\* l)

{

if(l == NULL || l->next == NULL) return l;

ListNode\* prev = NULL;

ListNode\* temp = NULL;

while(l != NULL)

{

temp = l->next;

l->next = prev;

prev = l;

l = temp;

}

return prev;

}

ListNode\* addTwoNumbers(ListNode\* l1, ListNode\* l2) {

ListNode\* revl1 = reverse(l1);

ListNode\* revl2 = reverse(l2);

int carry = 0, sum = 0;

ListNode\* add = NULL;

while(revl1 != NULL || revl2 != NULL)

{

sum = carry;

if(revl1 != NULL)

{

sum = sum + revl1->val;

revl1 = revl1->next;

}

if(revl2 != NULL)

{

sum = sum + revl2->val;

revl2 = revl2->next;

}

carry = sum/10;

sum = sum%10;

if(add == NULL)

{

add = new ListNode(sum);

}

else

{

add = new ListNode(sum, add);

}

}

if(carry != 0)

{

while(carry > 0)

{

int c = carry%10;

add = new ListNode(carry, add);

carry = carry/10;

}

}

return add;

}

};