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

***CODES BASED ON WEELKY TASK***

1. ***Swap Nodes in Pairs:***

class Solution {

public:

ListNode\* swapPairs(ListNode\* head) {

if(head == NULL) return head;

ListNode\* temp = head;

int value;

while(temp!=NULL && temp->next!=NULL)

{

value = temp->val;

temp->val = temp->next->val;

temp->next->val = value;

temp = temp->next->next;

}

return head;

}

};

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

class Solution {

public:

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

ListNode\* h1 = l1;

ListNode\* h2 = l2;

ListNode\* head = new ListNode();

ListNode\* temp = head;

bool c = true;

int carry = 0, sum = 0;

while(h1!=NULL || h2!=NULL)

{

sum = carry;

if(h1!=NULL)

{

sum = sum + h1->val;

h1 = h1->next;

}

if(h2!=NULL)

{

sum = sum + h2->val;

h2 = h2->next;

}

carry = sum/10;

sum = sum % 10;

if(head->val == 0 && head->next==NULL && c == true)

{

cout<<sum<<" ";

head = new ListNode(sum);

temp = head;

c = false;

}

else

{

temp->next = new ListNode(sum);

temp = temp->next;

}

}

if(carry == 1) temp->next = new ListNode(1);

return head;

}

};

1. ***Rotate List:***

class Solution {

public:

ListNode\* rotateRight(ListNode\* head, int k) {

if(head == NULL || k == 0) return head;

int i=0, j=0;

ListNode\* l1 = head;

ListNode\* l2 = head;

ListNode\* l3 = head;

while(l3!=NULL)

{

l3 = l3->next;

j++;

}

k = k % j;

if(k == 0) return head;

while(i<k)

{

l2 = head;

while(l2->next != NULL)

{

l1 = l2;

l2 = l2->next;

}

l2->next = head;

l1->next = NULL;

head = l2;

i++;

}

return head;

}

};