LinkList2 Assignment

Abhishek.21402@knit.ac.in

Q.1 You are given the head of a linked list. Delete the middle node, and return the head of the modified linked list. [Leetcode 2095].

```
class Solution {
public:
    ListNode* deleteMiddle(ListNode* head) {

if(!head or !head->next)return NULL;
ListNode *fast = head , *slow = head;
while(fast and fast->next){
slow = slow->next;
fast = fast->next->next;
}
ListNode *prev = NULL , *curr = head;
while(curr != slow){
prev = curr;
curr = curr->next;
```

```
}
prev->next = curr->next;
return head;
   }
};
```

Q.2 You are given two linked lists: list1 and list2 of sizes n and m respectively.

Remove list1 's nodes from the ath node to the bth node, and put list2 in their place.

[Leetcode 1669].

Code -

```
class Solution {
public:
   ListNode* mergeInBetween(ListNode* list1, int a, int b, ListNode* list2) {

       ListNode*temp1=list1;
        int len2=0;
       ListNode*temp2=list2;
       while(temp2->next!=NULL){
```

```
            len2++;
            temp2=temp2->next;
        }

        for(int i=1;i<a;i++){
            temp1=temp1->next;
        }
        ListNode*temp3=list1;
        for(int i=1;i<=b+1;i++){
            temp3=temp3->next;
        }
        temp1->next=list2;
        temp2->next=temp3;
        return list1;

    }
};
```

Q.3 You are given the head of a linked list, and an integer k .

Return the head of the linked list after swapping the values of the kth node from the beginning

and the kth node from the end (the list is 1-indexed). [Leetcode 1721].

Code- class Solution {

public:

```
   void swap(int k,int r,vector<int>&v){

      int temp=v[k];
      v[k]=v[r];
      v[r]=temp;
   }
   ListNode* swapNodes(ListNode* head, int k) {


     vector<int>v;
      ListNode*temp=head;
      int ln=0;
     while(temp){
```

```
            v.push_back(temp->val);
            temp=temp->next;
            ln++;
        }

        swap(k-1,ln-k,v);
        ListNode*temp2=head;
        int i=0;
        while(temp2){

            temp2->val=v[i];
            temp2=temp2->next;
            i++;
        }
        return head;
    }
};
```

Q.4 Given the head of a linked list and an integer val , remove all the nodes of the linked list that has

Node.val == val , and return the new head.

Code -

```
class Solution {
public:
   ListNode* removeElements(ListNode* head, int val) {
      ListNode*temp=head;
       while(temp and temp->val==val){
          temp=temp->next;
       }
       head=temp;
       while(temp){
          if(temp->next&&temp->next->val==val){
              temp->next=temp->next->next;
          }
          else
```

```
            temp=temp->next;
        }
        return head;
    }
};
```