NAME:SAURABH RAJ
REGISTERED EMAIL :saurabhraj25aug2004@gmail.com
COURSE NAME:DECODE DSA WITH C++
BATCH:DECODE 2.0
MODULE NAME:LINKED LIST Assignment part 2
MOBILE NUMBER:8434283953

QUESTION1:

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]**

   The **middle node** of a linked list of size `n` is the `⌊n / 2⌋th` node from the **start** using **0-based indexing**, where `⌊x⌋` denotes the largest integer less than or equal to `x`.

Answer:

```
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;
    }
```

Question:2

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]

Answer:

```
class Solution {
public:
    ListNode* mergeInBetween(ListNode* list1, int a, int b, ListNode* list2) {
        ListNode *curr = list1;
        a--;
        while(a--){
            curr = curr->next;
        }

        b++;
        ListNode *curr2 = list1;
        while(b--){
            curr2 = curr2->next;
        }

        ListNode *temp = list2;

        while(temp->next)temp = temp->next;
        temp->next = curr2;
        curr->next = list2;

        return list1;
    }
```

Question:3

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]**

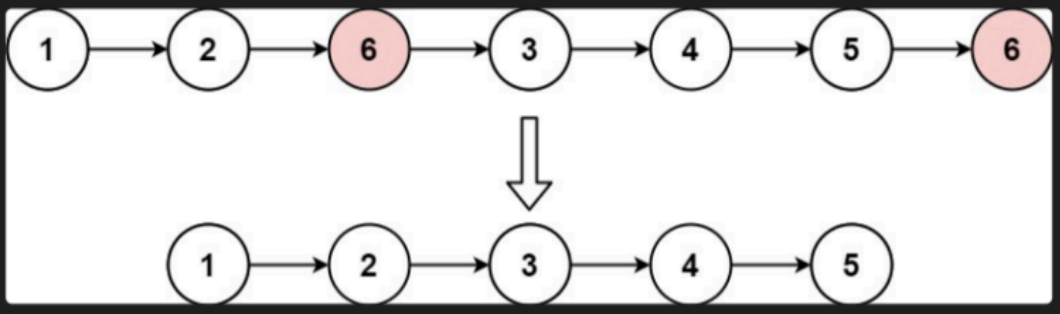
Answer:

```cpp
class Solution {
public:
    ListNode* swapNodes(ListNode* head, int k) {
        ListNode *temp = head;
        k--;
        while(k--)temp = temp->next;
        ListNode *p1 = temp->next , *p2 = head;

        while(p1){
            p1 = p1->next;
            p2 = p2->next;
        }

        swap(temp->val , p2->val);
        return head;
    }
};
```

Question:4

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*.

Example 1:



Answer:

```
class Solution {
public:
    ListNode* removeElements(ListNode* head, int val) {
        ListNode *curr = head;

        while(curr and curr->val == val){
            curr = curr->next;
        }

        head = curr;

        while(curr){
            if(curr->next and curr->next->val == val)
                curr->next = curr->next->next;
            else curr = curr->next;
        }
        return head;
    }
};
```

Question:5

5. Find the length of loop in Cycle of Linked List.

Answer:

```
#include<bits/stdc++.h>
using namespace std;
class node{
   public :
      int data;
      node *next;
      node(int n){
         data = n;
         next = NULL;
      }
};
class linkedlist{
   public:
      node *head,*tail;
      linkedlist(){
         head = NULL;
         tail = NULL;
      }
      void display(){
         node *temp = head;
```