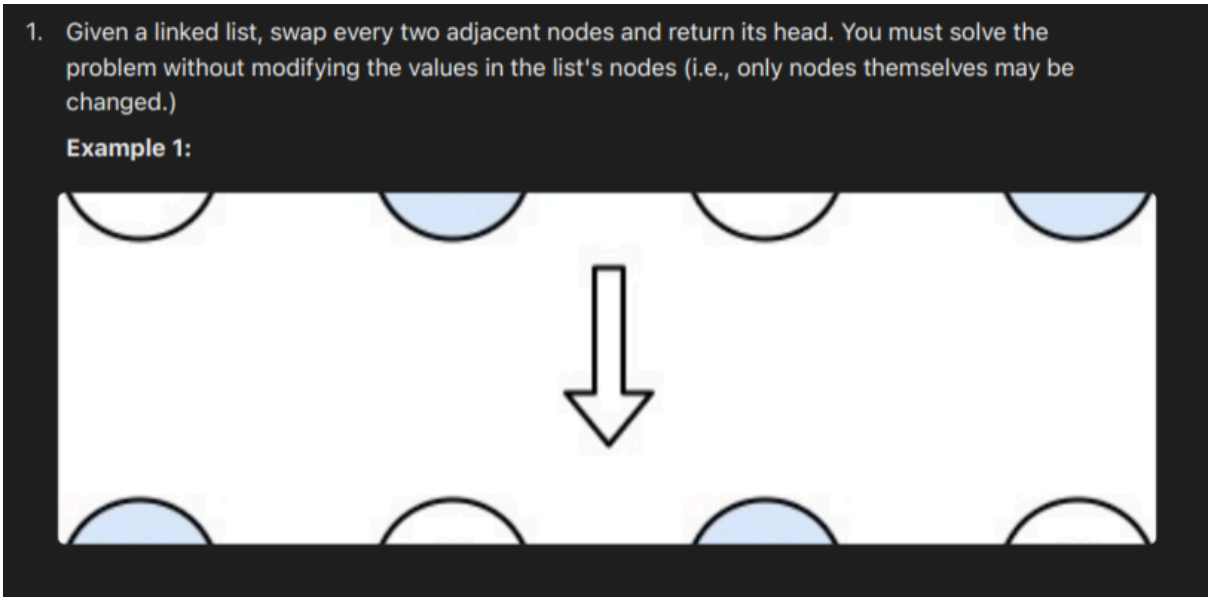
NAME:SAURABH RAJ
REGISTERED EMAIL :saurabhraj25aug2004@gmail.com
COURSE NAME:DECODE DSA WITH C++
BATCH:DECODE 2.0
MODULE NAME:LINKED LIST Assignment part 4
MOBILE NUMBER:8434283953

QUESTION1:



Answer:

```
class Solution {
public:
1
int length(ListNode *head){
int cnt = 0;
ListNode *temp = head;
while(temp){
cnt++;
temp=temp->next;
}
return cnt;
}
ListNode* reverseKGroup(ListNode* head, int k) {
int len = length(head);
```

```
cout<<len<<endl;
if(len < k or !head)return head;
ListNode *dummy = new ListNode(0);
dummy->next = head;
ListNode *curr = dummy;
ListNode *prev = dummy;
ListNode *nex = dummy;
while(len >= k){
curr = prev->next;
nex = curr->next;
for(int i=1;i<k;i++){
curr->next = nex->next;
nex->next = prev->next;
prev->next = nex;
nex = curr->next;
}
prev = curr;
len -= k;
}
return dummy->next;
}
ListNode* swapPairs(ListNode* head) {
return reverseKGroup(head , 2);
}
}
```

Question:2

2. You are given the `head` of a linked list, which contains a series of integers **separated** by `0`'s. The **beginning** and **end** of the linked list will have `Node.val == 0`.

Answer:

```
class Solution {
public:
 ListNode* mergeNodes(ListNode* head) {
 ListNode *dummy = new ListNode(0);
 dummy->next = head;
 ListNode *temp = dummy;
 int sum = 0;
 while(head){
 if(head->val == 0){
 temp->next = new ListNode(sum);
 temp = temp->next;
```

```
 ;
 sum = 0;
 }
 else{
 sum += head->val;
 }
 head = head->next;
 }
 return dummy->next->next;
 }
};
```