**Pairwise swap elements of a linked list: -**

**Easy** Accuracy: **52.06%** Submissions: **106K+** Points: **2**

Given a singly linked list of size **N**. The task is to swap elements in the linked list pairwise.  
For example, if the input list is 1 2 3 4, the resulting list after swaps will be 2 1 4 3.  
**Note**: You need to swap the nodes, not only the data. If only data is swapped then driver will print -1.

**Example 1:**

**Input:**

LinkedList: 1->2->2->4->5->6->7->8

**Output:**2 1 4 2 6 5 8 7

**Explanation:**After swapping each pair considering (1,2), (2, 4), (5, 6).. so on as pairs, we get 2, 1, 4, 2, 6, 5, 8, 7 as a new linked list.

**Example 2:**

**Input:**

LinkedList: 1->3->4->7->9->10->1

**Output:**3 1 7 4 10 9 1

**Explanation:**After swapping each pair considering (1,3), (4, 7), (9, 10).. so on as pairs, we get 3, 1, 7, 4, 10, 9, 1 as a new linked list.

**Your Task:**  
The task is to complete the function **pairWiseSwap**() which takes the head node as the only argument and returns the head of modified linked list.

**Expected Time Complexity:**O(N).  
**Expected Auxiliary Space:**O(1).

**Constraints:**  
1 ≤ N ≤ 105

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int data;

struct Node\* next;

Node(int x){

data = x;

next = NULL;

}

};

// } Driver Code Ends

/\*

Pairwise swap a linked list

The input list will have at least one element

node is defined as

struct Node

{

int data;

struct Node\* next;

Node(int x){

data = x;

next = NULL;

}

}\*head;

\*/

class Solution

{

public:

Node\* pairWiseSwap(struct Node\* head)

{

if (!head) return head;

if (!head->next) return head;

Node \*ptr = head;

Node \*ret = ptr->next;

while (ptr && ptr->next) {

Node \*tmp1 = ptr;

Node \*tmp2 = ptr->next;

ptr = ptr->next->next;

tmp2->next = tmp1;

tmp1->next = (ptr && ptr->next) ? ptr->next : ptr;

}

return ret;

}

};

//{ Driver Code Starts.

void printList(Node\* node)

{

while (node != NULL) {

cout << node->data <<" ";

node = node->next;

}

cout<<"\n";

}

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int data;

cin>>data;

struct Node \*head = new Node(data);

struct Node \*tail = head;

map<Node\*, int> mp;

mp[head] = head->data;

for (int i = 0; i<n-1; ++i)

{

cin>>data;

tail->next = new Node(data);

tail = tail->next;

mp[tail] = tail->data;

}

struct Node \*failure = new Node(-1);

Solution ob;

head = ob.pairWiseSwap(head);

int flag = 0;

struct Node \*temp = head;

while(temp){

if(mp[temp] != temp->data){

flag = 1;

break;

}

temp = temp->next;

}

if(flag)

printList(failure);

else

printList(head);

}

return 0;

}

// } Driver Code Ends

**T.C: - O(N)**

**S.C: - O(1)**