**Problem Statement**

**Title: Swap Nodes in Pairs**

**Problem Description:** Mia is organizing pairs of photos for a slideshow. She wants to swap every two adjacent nodes in her linked list. Given a linked list, swap every two adjacent nodes and print its head. You must solve the problem without modifying the values in the list's nodes (i.e., only nodes themselves may be changed).

**Input Format:**

* The first line contains an integer n, the number of elements in LinkedListA.
* The second line contains n space-separated elements of LinkedListA.

**Output Format:** Print the linked list after swapping nodes in pairs.

**Constraints:**

* The number of nodes in the linked list is in the range [0, 1000].
* The elements of the linked list are integers.

**Examples:**

**Example 1:**

Input:

4

1 2 3 4

Output:

2 1 4 3

**Example 2:**

Input:

3

1 2 3

Output:

2 1 3

**Test Cases:**

**Test Case 1:**

Input:

5

1 2 3 4 5

Output:

2 1 4 3 5

**Test Case 2:**

Input:

6

10 20 30 40 50 60

Output:

20 10 40 30 60 50

**Test Case 3:**

Input:

4

5 10 15 20

Output:

10 5 20 15

**Test Case 4:**

Input:

2

8 9

Output:

9 8

**Test Case 5:**

Input:

1

7

Output:

7

**Solution in Python**

Here is a Python solution that swaps every two adjacent nodes in the linked list:

python

Copy code

class ListNode:

def \_\_init\_\_(self, x):

self.val = x

self.next = None

def swap\_pairs(head):

dummy = ListNode(0)

dummy.next = head

current = dummy

while current.next and current.next.next:

first = current.next

second = current.next.next

first.next = second.next

current.next = second

current.next.next = first

current = current.next.next

return dummy.next

# Helper function to create a linked list from a list

def create\_linked\_list(lst):

if not lst:

return None

head = ListNode(lst[0])

current = head

for value in lst[1:]:

current.next = ListNode(value)

current = current.next

return head

# Helper function to print the linked list

def print\_linked\_list(head):

current = head

result = []

while current:

result.append(str(current.val))

current = current.next

return " ".join(result)

def main():

import sys

input = sys.stdin.read

data = input().strip().split()

n = int(data[0])

listA = list(map(int, data[1:n+1]))

headA = create\_linked\_list(listA)

swapped\_head = swap\_pairs(headA)

print(print\_linked\_list(swapped\_head))

if \_\_name\_\_ == "\_\_main\_\_":

main()

This solution reads the input values, creates the linked list, and then swaps every two adjacent nodes using the swap\_pairs function. The output is the modified linked list printed in the required format.