NAME :- PIYUSH VERMA || Registration no :- 23MCA1104

You are given the head of a singly linked-list. The list can be represented as:

L0 → L1 → … → Ln - 1 → Ln
Reorder the list to be on the following form:

L0 → Ln → L1 → Ln - 1 → L2 → Ln - 2 → …
You may not modify the values in the list's nodes. Only nodes themselves may be changed.

Example:
Input: head = [1,2,3,4]
Output: [1,4,2,3]

Input: head = [1,2,3,4,5]
Output: [1,5,2,4,3]

CODE

```
#include <stdio.h>
#include <stdlib.h>

struct ListNode {
    int val;
    struct ListNode *next;
};

// Function to reverse a linked list
struct ListNode* reverseList(struct ListNode* head) {
    struct ListNode* prev = NULL;
    struct ListNode* current = head;

    while (current != NULL) {
        struct ListNode* nextTemp = current->next;
        current->next = prev;
        prev = current;
        current = nextTemp;
    }

    return prev;
}
```

```
// Function to reorder the linked list
void reorderList(struct ListNode* head) {
    if (head == NULL || head->next == NULL || head->next->next == NULL) {
        return;
    }

    // Find the middle of the linked list
    struct ListNode* slow = head;
    struct ListNode* fast = head;
    while (fast->next != NULL && fast->next->next != NULL) {
        slow = slow->next;
        fast = fast->next->next;
    }

    // Reverse the second half of the linked list
    struct ListNode* secondHalf = reverseList(slow->next);
    slow->next = NULL;

    // Merge the first and reversed second halves alternatively
    struct ListNode* current1 = head;
    struct ListNode* current2 = secondHalf;
    while (current2 != NULL) {
        struct ListNode* next1 = current1->next;
        struct ListNode* next2 = current2->next;

        current1->next = current2;
        current2->next = next1;

        current1 = next1;
        current2 = next2;
    }
}

// Function to print the linked list
void printList(struct ListNode* head) {
    struct ListNode* current = head;
    while (current != NULL) {
        printf("%d -> ", current->val);
        current = current->next;
    }
    printf("NULL\n");
}

int main() {
    // Example usage
    struct ListNode* head = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->val = 1;
```

```
    head->next = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->next->val = 2;
    head->next->next = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->next->next->val = 3;
    head->next->next->next = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->next->next->next->val = 4;
    head->next->next->next->next = NULL;

    printf("Original linked list: ");
    printList(head);

    reorderList(head);

    printf("Reordered linked list: ");
    printList(head);

    // Free memory
    while (head != NULL) {
        struct ListNode* temp = head;
        head = head->next;
        free(temp);
    }

    return 0;
}
```

OUTPUT