|  |  |
| --- | --- |
| **Pair Wise swap in C++** | |
| #include <iostream>  using namespace std;  // Node class definition  class Node {  public:  int data;  Node\* next;  Node(int d) {  data = d;  next = nullptr;  }  };  // PairwiseSwapLL class definition  class PairwiseSwapLL {  public:  Node\* head;  PairwiseSwapLL() {  head = nullptr;  }  // Method to print the elements of the list  void printList(Node\* node) {  while (node != nullptr) {  cout << node->data << " ";  node = node->next;  }  cout << endl;  }  // Method to perform pairwise swapping of nodes  Node\* pairWiseSwap(Node\* node) {  if (node == nullptr || node->next == nullptr) {  return node;  }  Node\* remaining = node->next->next;  Node\* newHead = node->next;  node->next->next = node;  node->next = pairWiseSwap(remaining);  return newHead;  }  };  int main() {  // Create an instance of PairwiseSwapLL  PairwiseSwapLL list;  // Construct the linked list: 1->2->3->4->5->6->7  list.head = new Node(1);  list.head->next = new Node(2);  list.head->next->next = new Node(3);  list.head->next->next->next = new Node(4);  list.head->next->next->next->next = new Node(5);  list.head->next->next->next->next->next = new Node(6);  list.head->next->next->next->next->next->next = new Node(7);  // Display the original list  cout << "Linked list before calling pairwiseSwap() " << endl;  list.printList(list.head);  // Perform pairwise swapping  list.head = list.pairWiseSwap(list.head);  // Display the list after pairwise swapping  cout << "Linked list after calling pairwiseSwap() " << endl;  list.printList(list.head);  // Clean up allocated memory  Node\* curr = list.head;  Node\* next = nullptr;  while (curr != nullptr) {  next = curr->next;  delete curr;  curr = next;  }  return 0;  } | **Dry Run Table**  Input List: 1 → 2 → 3 → 4 → 5 → 6 → 7   | **Recursive Call** | **node** | **Swapped Pair** | **Remaining** | **Result after call** | | --- | --- | --- | --- | --- | | 1 | 1 | 1 ↔ 2 | 3 | 2 → 1 → ? | | 2 | 3 | 3 ↔ 4 | 5 | 4 → 3 → ? | | 3 | 5 | 5 ↔ 6 | 7 | 6 → 5 → ? | | 4 | 7 | no pair | nullptr | 7 |   🔁 Backtracking:   * 4th call returns: 7 * 3rd call builds: 6 → 5 → 7 * 2nd call builds: 4 → 3 → 6 → 5 → 7 * 1st call builds: 2 → 1 → 4 → 3 → 6 → 5 → 7   ✅ **Final Output:**  2 1 4 3 6 5 7 |
| Linked list before calling pairwiseSwap()  1 2 3 4 5 6 7  Linked list after calling pairwiseSwap()  2 1 4 3 6 5 7 | |