Data Structures

Name: [redacted]

Quiz 1

Roll No: [redacted]

Question: Following is the code for reversing a doubly linked list after k nodes. Dry Run this code and write the output first. **(3+7)**

Output:

1 2 3 4 5 6

1 2 3 4 5 6

I guess this is not working as it should be working... So, identify that and add your additional suggested code which should rectify the mistake that you encountered ... (Please note that you are not allowed to write the function from scratch or remove anything. Just identify the missing part in this function and your code to make it work right)

For your additional code chunk in the function:

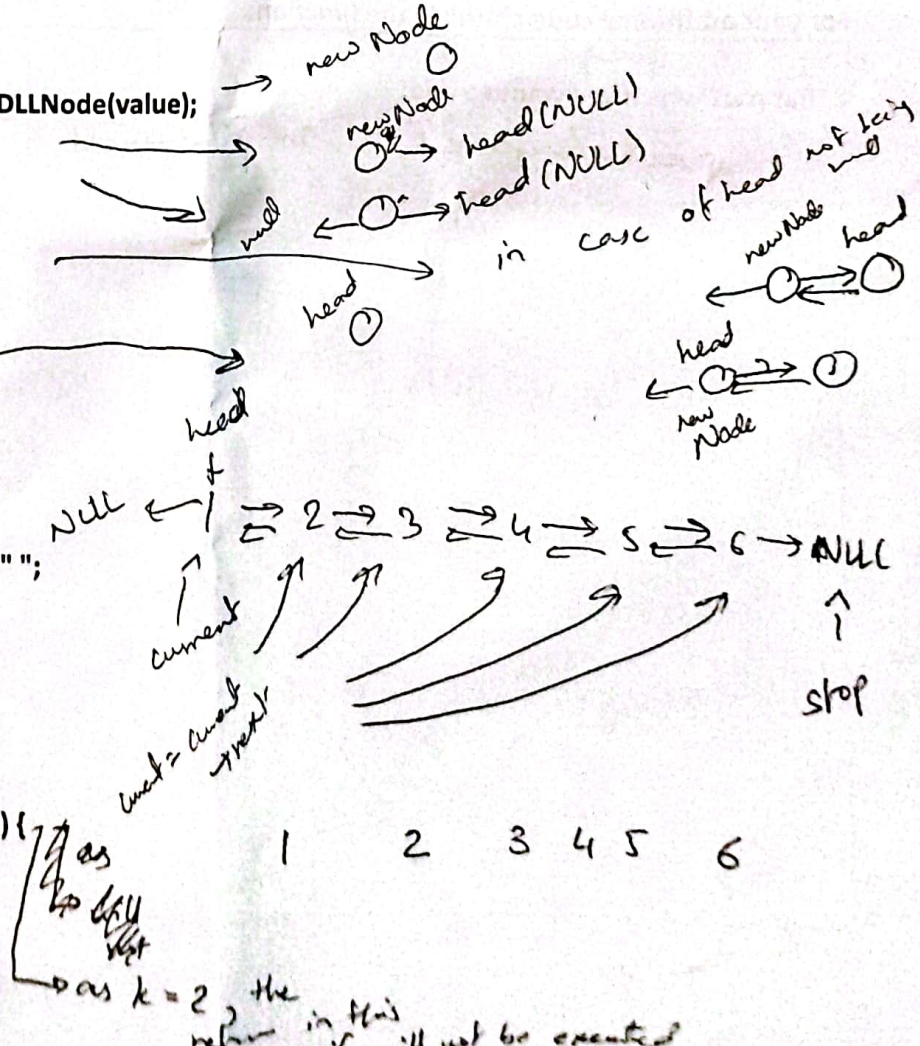
That part which you want to add:

previousGroupEnd = groupEnd->next;

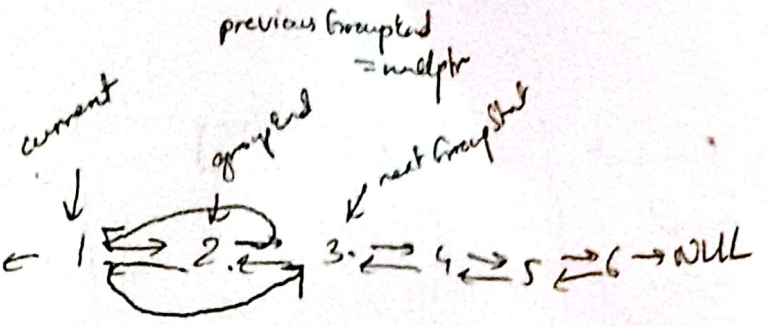
Your output (after) should be:

2<->1<->4<->3<->6<->5<->nullptr

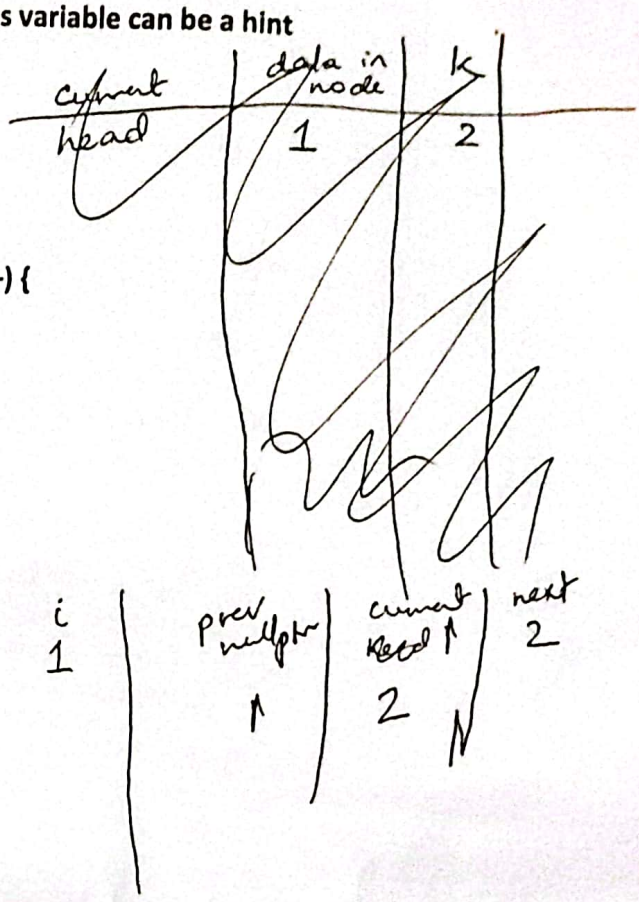
```
#include<iostream>
using namespace std;

template <class T>
class DLList {
private:
   struct DLLNode {
      T data;
      DLLNode* prev;
      DLLNode* next;
      DLLNode(T value) : data(value), prev(nullptr), next(nullptr) {}
   };

   DLLNode* head;

public:
   DLList() : head(nullptr) {}

   void insertAtStart(T value) {
      DLLNode* newNode = new DLLNode(value);
      newNode->next = head;
      newNode->prev = nullptr;

      if (head != nullptr) {
         head->prev = newNode;
      }
      head = newNode;
   }

   void print() const {
      DLLNode* current = head;
      while (current != nullptr) {
         cout << current->data << " ";
         current = current->next;
      }
      cout << endl;
   }

   void reverseInGroups(int k) {
      if (head == nullptr || k <= 1) {
         return;
      }
```



k = 2




```
DLLNode* current = head;
DLLNode* previousGroupEnd = nullptr;          //this variable can be a hint

while (current != nullptr) {
  DLLNode* groupStart = current;
  DLLNode* groupEnd = groupStart;


  for (int i = 1; i < k && groupEnd->next != nullptr; i++) {
    groupEnd = groupEnd->next;
  }

  DLLNode* nextGroupStart = groupEnd->next;

  DLLNode* prev = nullptr;
  current = groupStart;

  while (current != nextGroupStart) {
    DLLNode* next = current->next;
    current->next = prev;
    current->prev = next;
    prev = current;
    current = next;
  }
  //complete the missing code here
 }
}

};
```




```
int main() {
  DLList<int> DLL1;
  DLL1.insertAtStart(6);
  DLL1.insertAtStart(5);
  DLL1.insertAtStart(4);
  DLL1.insertAtStart(3);
  DLL1.insertAtStart(2);
  DLL1.insertAtStart(1);
  DLL1.print();
  cout << endl;
  DLL1.reverseInGroups(2);
  DLL1.print();
}
```