Class LLNode representing a node of singly linked lists is declared as below:

class LLNode {

public:

    int val;

    LLNode\* next;

    LLNode(); // Constructor: val = 0, next = nullptr

    LLNode(int val, LLNode\* next); // Constructor with customized data

}

Given a singly linked list head node.

Your task is to implement a function with following prototype:

LLNode\* reverseLinkedList(LLNode\* head);

The function returns head node of the reversed singly linked list.

**Note:**

- The iostream library has been included and namespace std is being used. No other libraries are allowed.

- The constructors and methods of class LLNode have been defined fully so you do not need to redefine them.

- You can write helper functions.

**For example:**

| **Test** | **Result** |
| --- | --- |
| int arr[] = {13, 88, 60, 7, 192};  LLNode\* head = LLNode::createWithIterators(arr, arr + sizeof(arr) / sizeof(int));  LLNode::printList(head);  cout << "\n";  LLNode\* newhead = reverseLinkedList(head);  LLNode::printList(newhead);  newhead->clear(); | [13, 88, 60, 7, 192]  [192, 7, 60, 88, 13] |