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 and a integer k.

Your task is to implement a function with following prototype:

LLNode\* rotateLinkedList(LLNode\* head, int k);

The function returns head node of the rotated singly linked list obtained after rotate the linked list to the right by k places.

**Note:**

- The iostream library has been used 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[] = {2, 4, 6, 6, 3};  int k = 3;  LLNode\* head = LLNode::createWithIterators(arr, arr + sizeof(arr) / sizeof(int));  LLNode::printList(head);  cout << "\n";  LLNode\* newhead = rotateLinkedList(head, k);  LLNode::printList(newhead); | [2, 4, 6, 6, 3]  [6, 6, 3, 2, 4] |