

Reverse linked list - 2. (LC-92)

`ListNode* reverseBetween (ListNode* head, int left, int right)`

`if (!head || left == right) return head;`

`ListNode* dummy(0)`

`dummy → next = head`

`ListNode* prev = &dummy.`

`for (int i=1; i<left; i++) {`

`prev = prev → next`

`ListNode* curr = prev → next`

`ListNode* next = NULL`

`ListNode* beforeLeft = prev`

`// reverse the nodes starting from left upto right:`

`for (int i = left; i < right; i++) {`

`next = curr → next`

`curr → next = next → next`

`next → next = beforeLeft → next`

`beforeLeft → next = next`

`}`

`return dummy.next;`