REVERSING A DLL

☆ Given the head of a doubly linked list, we are supposed to return the head of its reversed version

Brute force solution is to swap the data of each node with its respective. We can take a stack or any other data structure to store all the elements. Then we can just iterate backwards and place those elements again. Time complexity is O(2N) and space complexity is O(N).

Optimally, we can reverse the links for each node and return the last link as the new head.

```
Node * reverseDLL(Node * head) {
    Node * current = head;
    Node * last = nullptr;
    while(current != nullptr) {
        last = current -> back;
        current -> back = current -> next;
        current -> next = last;
        current = current -> back;
    }
    return last -> back;
```