Problem 2: Operations on Singly Linked List

Data Structures Lab (CS111)

Consider a singly linked list S on which following operations are allowed:

1. $insert\_first(x)$: This operation inserts element x at the start of the list.
2. $delete\_first()$: This operation deletes the first element of the list and returns that element.
3. $insert\_last(x)$: This operation inserts element x at the end of the list.
4. $delete\_last()$: This operation deletes the last element of the list and returns that element.

Write C functions to implement all four operations mentioned above. Build S where user enters the element of S using these operations only. Then, implement following operations using these functions only:

1. $swap\_ends(S)$: Swap the first and last items in S.
2. $shift\_left(S, k)$: Move the first k items in order to the start of the list S. (After, this operation k^{th} item should be last and the $(k+1)^{st}$ item should be first element of S. This is essentially the circular left shift)

Hint: You can maintain a tail pointer along with the head pointer while building the list.
Note: You cannot use any other operations like $traverse\_list(S)$, manual swap of $node->data$, etc.