

12438 - DS_2019Fall_Linked List (practice)

[Status \(/status/?pid=12438\)](/status/?pid=12438) | Limits[Submit \(/users/submit/12438\)](/users/submit/12438)

Description

Implement some operations used in a linked list

ex. insert, delete, reverse, swap, rotate, print

1. **InsertBack data:** insert a node with data to the end of the linked list.
2. **InsertFront data:** insert a node with data to the front of the linked list.
3. **InsertAfter data ref:** insert a node with data after the existing node with ref in the linked list.
4. **InsertBefore data ref:** insert a node with data before the existing node with ref in the linked list.
5. **Delete data:** delete the node with data (you may need to free the memory).
Please note that our input will ensure that each node holds a distinct integer (data field).
6. **DeleteFront:** delete the first node in the linked list.
7. **DeleteBack:** delete the last node in the linked list.
8. **Reverse:** reverse the nodes order of the linked list.
9. **Rotate k:** put the last k nodes into the front (if $k > \text{total}$ nodes, then $k = k \% \text{total}$).
10. **Swap data1 data2:** swap the positions of node holding data1 and data2 (if one node doesn't exist, do nothing).
Please note that our input will ensure that each node holds a distinct integer (data field).
11. **PrintChain:** print the data from the first node to the last node in the form of data1->data2. (we provide an unfinished code and PrintChain has already been implemented).

Input

Each testcase will contain N operations, $0 < N < 100000$.

All integer data are between 0 to 100000.

There are no two nodes with the same data in this problem. That is, we will not have testcase like InsertFront 1, InsertFront 1.

Output

For output operation PrintChain, the form should be 1->2->3->4. Integers (data fields in nodes) are separated by '->'.

Sample Input

Download (data:text/plain;charset=utf-8,DeleteBack%0D%0AInsertBack%200%0D%0AInsertBack%201%0D%0AInsertBack%202%0D%0AInsertBack%203%0D%0AInsertBack%204%0D%0APrintChain%0D%0ADelete%202%0D%0ASwap%201%204%0D%0ARotate%202%0D%0APrintChain%0D%0AReverse%0D%0APrintChain%0D%0A)

```
DeleteBack
InsertBack 0
InsertBack 1
InsertBack 2
InsertBack 3
InsertBack 4
PrintChain
Delete 2
Swap 1 4
Rotate 2
PrintChain
Reverse
PrintChain
```

Sample Output

Download (data:text/plain;charset=utf-8,0-%3E1-%3E2-%3E3-%3E4%0D%0A3-%3E1-%3E0-%3E4%0D%0A4-%3E0-%3E1-%3E3%0D%0A)

```
0->1->2->3->4
3->1->0->4
4->0->1->3
```

Tags

oh

Discuss