Lab 7 Linked List

For this problem, you need to know how to implement linked lists.

You must implement the seven operations.

- void Push_back(int x) : Insert a node to the end of the linked list, the node's value is x.
  - Ex:
    List1: 6 => 2 => 7 => 4 => null
    List1.Push_back(12)
    List1: 6 => 2 => 7 => 4 => 12 => null

- void Push_front(int x) : Insert a node to the front of the linked list, the node's value is x.
  - Ex:
    List1: 6 => 2 => 7 => 4 => null
    List1.Push_ front(19)
    List1: 19 => 6 => 2 => 7 => 4 => null

- void Insert(int index,int x) : Insert a node to the linked list at position "index", the node's value is x.
  Note: The index of the first node in the linked list is 0. The index must be smaller than the length of the linked list.
  - Ex:
    List1: 6 => 2 => 7 => 4 => null
    List1. Insert(1,19)
    List1: 6 => 19 => 2 => 7 => 4 => null
    List1. Insert(0,39)
    List1: 39 => 6 => 19 => 2 => 7 => 4 => null
    List1. Insert(6,56)
    List1: 39 => 6 => 19 => 2 => 7 => 4 => 56 => null

- void Delete(int index) : Remove the node with index "index" in the linked list.
  - Ex:
    List1: 19 => 31 => 23 => 41 => 53 => null
    List1.Delete(3)
    List1: 19 => 31 => 23 => 53 => null
    List1.Delete(0)
    List1: 31 => 23 => 53 => null

- void Reverse() : Reverse the linked list.
  - Ex:
    List1: 1 => 3 => 5 => 7 => 9 => null
    List1. Reverse()
    List1: 9 => 7 => 5 => 3 => 1 => null

- void Swap(int index1, int index2) : Swap the two nodes which are at index1 and index2 respectively in the linked list.
  Note: The index of the first node in the linked list is 0. The index must be smaller than the length of the linked list.
  - Ex:
    List1: 1 => 3 => 5 => 7 => 9 => null
    List1. Swap(0,1)
    List1: 3 => 1 => 5 => 7 => 9 => null
    List1. Swap(2,4)
    List1: 3 => 1 => 9 => 7 => 5 => null
    List1. Swap(3,3)
    List1: 3 => 1 => 9 => 7 => 5 => null

- void Print() : Print all the elements in the linked list in order.
  - Ex:
    List1: 9 => 7 => 5 => 3 => 1 => null
    List1.Print()
    List: 9 7 5 3 1

You can assume that Insert() and Delete() will only do legal calculations.

You must use the template to do this lab.

Input Format

Please implement the file I/O part.
You MUST read the input data from the input1.txt.

The first line shows the number of test cases.

Each of the following lines:
The first number represents numbers of operations.
After the first number, there will be a character representing the operation(b,f,I,d,r,s).
“ b ” : represent Push_back(…)
“ f ” : represent Push_front(…)

There is a number after ‘ b ’ and ‘ f ’ , and that number we want to add to the list.

“ i ” : represent Insert(…)
There are two numbers after ‘ i ’, first number is index represents location we want to add node to list, second number is node’s value we want to add to list.

“ d ”: represent Delete(…)
There is a number after ‘ d ’ , this number represents the location we want to remove the node from list.

“ r ” : represent Reverse(…)
There is no number after ‘ d ’.

“ s ” : represent Swap(…)
There are two numbers after ‘ s ’, the first number is the first index, the second number is the second index.

Output Format

You must print all the content of the linked list in order after doing each calculation. See more detail from Sample output.

Sample Input & Output.

Input:

5
6 f 1 f 2 b 3 b 4 i 2 5 r
6 f 1 f 2 f 3 f 4 f 5 r
7 f 1 f 2 f 3 f 5 f 10 d 0 s 0 3
8 b 1 f 2 b 3 i 1 5 d 2 r s 0 1 r
13 b 1 f 2 b 3 f 4 i 1 5 i 2 6 i 3 7 r d 0 r d 0 r d 0

Output:

List: 4 3 5 1 2

List: 1 2 3 4 5

List: 1 3 2 5

List: 2 3 5

List: 2 7 6 5