Q6. Linked List

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

You must implement the six operation.

- 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.
  - 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 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 template to do this lab.

Input Format

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

The first line shows the number of test cases.

Each of the following lines:
The first number represents numbers of operations.
After first number, there will be a character representing the operation(b,f,i,d,r).
“ b ” : represent Push_back(…)
“ f ” : represent Push_front(…)
There is a number after ‘ b ’ and ‘ f ’ , and that number we want to add to 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 location we want to remove the node from list.

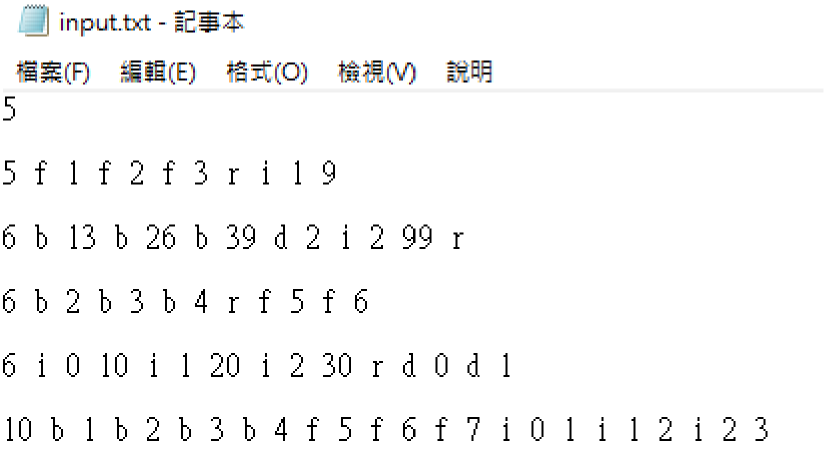
“ r ” : represent Reverse(…)
There is no number after ‘ r ’.

Output Format

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

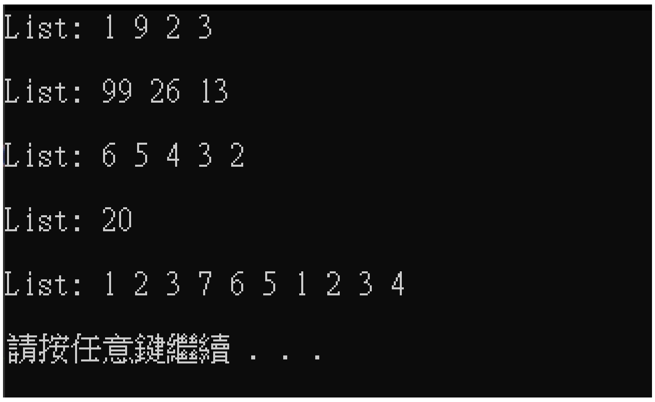
Sample Input & Output.

Input:



```
5
5 f 1 f 2 f 3 r i 1 9
6 b 13 b 26 b 39 d 2 i 2 99 r
6 b 2 b 3 b 4 r f 5 f 6
6 i 0 10 i 1 20 i 2 30 r d 0 d 1
10 b 1 b 2 b 3 b 4 f 5 f 6 f 7 i 0 1 i 1 2 i 2 3
```

Output:



```
List: 1 9 2 3
List: 99 26 13
List: 6 5 4 3 2
List: 20
List: 1 2 3 7 6 5 1 2 3 4
請按任意鍵繼續 . . .
```