Question 2

Write a program which creates a doubly linked list of *integers* and then sorts that doubly linked list in-place. 'In-place' sorting means that the program should not use any extra memory in form of array or list. In other words, an in-place sorting algorithm sorts the given input using O(1) additional storage space.

Input:
Since you're expected to implement a linked list, initial number of elements are not specified. Your program should take input until the user enters 'stop 0' or 'stop 1' (i.e. stop zero or stop one, without the quotes).
Each line of input could be of one of the following types:

➔ insertbeg K
Insert a number K to the beginning of the list

➔ insertend K
Insert a number K to the end of the list

➔ deletebeg
Delete the node from the beginning of the list. If the list is empty, do nothing.

➔ deletelast
Delete the last node. If the list is empty, do nothing.

➔ stop 0
Termination of the input file

➔ stop 1
Termination of the input file

Output:
On encountering a 'stop 0' (stop zero), the program shall sort the doubly linked list and print the entire list (from head to tail). Each node of the list shall be printed in a new line.
On encountering a 'stop 1' (stop one), the program shall sort the doubly linked list and print the entire list in **reverse** order (from tail to head) . Each node of the list shall be printed in a new line.

Constraints:
1 ≤ K ≤ 10^9

Sample Input:
insertbeg 8
insertbeg 10
insertend 12
deletebeg
insertend 7
insertbeg 9
deletelast
insertend 5

stop 0

Sample Output:

```
5
8
9
12
```

Explanation:

The list entered by the user is as: 9 <-> 8 <-> 12 <-> 5. It's sorted order will be as given by the sample output.