CSE 4304: Data Structures
Lab: 06
Topic: Problems related to Linked Lists

Task 1

Implement Insertion operation using ***Singly-Linked List***. The user might want to insert an integer **v** in the beginning /at the end / after any particular element.
First line will be the number of operations **N**. Followed by **N** operations in the format as:

- 1 v: insert the value **v** at the beginning of the Linked list.
- 2 v: insert **v** at the end of the list.
- 3 v key: insert **v** after the node having value **key** in the list.

After each insertion, show the status of the linked list. Write three different functions for these three types of insertion. Use a separate function to print the elements of the linked list.
Pass all the necessary elements using the parameter list. Do not use any Global variable.

Sample Input	**Sample Output**
7 1 15 1 17 2 22 3 15 99 1 77 3 33 44 2 55	15 17 15 17 15 22 17 15 99 22 77 17 15 99 22 Value not found. 77 17 15 99 22 77 17 15 99 22 55

Task 2

Implement ***Deletion*** operation using ***Singly-Linked List***. The user might want to Delete an integer from the beginning/end/any particular element.

First line will be the number of operations **N**. Followed by **N** operations in the format as:

- 1 : delete a value from the beginning of the Linked list.
- 2 : delete a value from the end of the list.
- 3 key: delete the node having the **key** from the list (if present).

Use the insertion function from **Task 1**. Insert the initial values as (10, 20, 30, 40, 50, 60, 70)
After each deletion, show the status of the linked list. Write three different functions for these three types of deletion. Use a separate function to print the elements of the linked list.
Pass all the necessary elements using the parameter list. Do not use any Global variable.

Sample Input	**Sample Output**
5 1 2 3 50 3 55 2	10 20 30 40 50 60 70 (initial value) 20 30 40 50 60 70 20 30 40 50 60 20 30 40 60 Value not present 20 30 40 60 20 30 40

Task 3,4:

Implement **Insertion** and **Deletion** operation using ***Doubly Linked Lists*** following the instructions of Task 1,2. (Each node will now have three attributes: data, next & previous.)