

# Assignment No: 01

## Data Structure and Algorithms

August 23, 2023

### 1 Retrieving The Trump

A local restaurant wants to chose a lucky winner and the policy they decided is, they will pick the middle customer of the day as winner. If the number of customers is even, then they will chose the middle two.

#### Input Format

1. The input will be read from a **.txt** file.
2. There will be three commands: ADD [number], DELETE, END.
3. The ADD command will be followed by an integer which will be appended at the **end** of the linked list, while DELETE command will remove the **last** element of the linked list. The END command will indicate EOF(End-of-file).

#### Example

Consider the following example:

```
ADD 24
ADD 10
ADD 65
ADD 98
DELETE
END
```

Now the linked list will be:

24 -> 10 -> 65 -> null

## Constraints

1. The ADD will always be followed by an integer.
2. There will be no **duplicate** entries of users.
3. Your **GetTrump()** method must always **print** the middle number in odd cases. In even cases, the method must **print** the middle two numbers.
4. Your **GetTrump()** method must always return the middle number(s) immediately **without the need of traversing** the whole linked list when the **GetTrump()** method is called.

## 2 Survival of the Fittest

A company by the name Kronstadt Industries is facing a tough competition from its rivals. As being one of the leaders of the market, Kronstadt Industries now wants to upgrade their algorithm so that they can overtake their fierce rivals.

The algorithm requires to calculate the median of a running stream of numbers. In easy terms, the count of numbers can't be determined at any stage. You have to code such an algorithm using Linked List instead of arrays, that can seamlessly handle running stream of numbers and can maintain the median of them as well.

### Input Format

1. The input will be read from a **.txt** file.

### Example

Consider the following example:

7 *Median : 7*

10 *Median : 8.5*

3 *Median : 7*

33 *Median : 8.5* and so on...

98

34

56

2

With each addition of number, the median of the stream must be updated as well.

## Constraints

1. The numbers will always be integers.
2. Numbers can contain duplicates.
3. Every individual number will always be greater than 0 and less than or equal to 100.

## 3 Reversing the Irreversible

The Cyber Security department of the Sanguin bank has observed some fraudulent transactions in one of their Gold customer's account. They are in desperate need to reverse those transactions as it will hurt the reputation of the bank.

Sanguin has signed you in to design such a system where a fraudulent transaction can be reversed easily.

## Input Format

1. The input will be read from a **.txt** file.
2. The first line contains the **numberOfTransactions** —space— **numberOfFraudulentTransactions** .
3. The lines afterwards represents the transactions in the format **transactionId** —space— **transactionAmount**.
4. After the number of transactions, the fraudulent transaction ids are mentioned.

## Example

Consider the following example:

```
5 2
01203,10000
12389,76500
9836,170800
1287,89100
8763,43000
9836
12389
```

So the transactions with these ids will be deleted from the Linked List.

### Constraints

1. Transactions ids and amounts will always be integer.
2. Transaction ids will always be unique.
3. Node with the transaction id will always exist.

## 4 The Editor (Difficulty: Hard)

Create an editor with a cursor that supports multiple functions.

### Commands

Commands the editor must support:

1. Add number: Add the number at the end of list.
2. Delete: Delete the tail of the list.
3. Add index number: Add the number at the specific index.
4. Delete index: Delete the specific index of the list.
5. MoveForward steps: Move the cursor forward by the number of steps mentioned.
6. MoveBackward steps: Move the cursor backwards by the number of steps mentioned.
7. AddImmediate number: Add the number just after the cursor position.
8. DeleteImmediate: Delete the index where the cursor is presented.
9. Print: Print the data of the cursor's node.
10. Update index number: Update the data at the specific index by the number.
11. UpdateImmediate number: Update the data at which the cursor is pointing to.
12. Shift index: Shift the current data of cursor with the data of the index specified.
13. Undo: Undo the last performed operation.

### Input Format

The input will be read from a **.txt** file.

### Constraints

1. The data of the list will be integer.
2. Operations can be invalid.
3. Undo will only support five recent operations.
4. Cursor will always start at the 0th index.

### Hints

1. Try to implement one functionality at a time.
2. UNDO can be performed via stack. You can use built-in stack of C++, although implementing stack yourself will yield additional points in final grading.
3. Try to break the code into multiple functions.
4. Try to have proper error handling as user input can be invalid.

## 5 As the Matter of Third

Emma has just learnt linked list, but she is stuck at finding the third last element of the linked list. You have to help Emma to find the third last element of linked list.

### Input Format

The input will be read from a **.txt** file.

### Example

Consider the following example:

78  
12  
43  
2  
12  
98

Your function will return the third last element, in this case, it will be 2.

### **Constraints**

1. The numbers will be integers.
2. All the numbers will be greater than 0.
3. If the link list's length is less than three, then -1 will be returned.