**1. Node Class**

* Represents a node in a doubly linked list where each node stores a binary bit.
* Contains **bit** to store the binary value, **next\_node** and **prev\_node** pointers for the next and previous nodes.

**2. List Class**

* Manages a list of binary bits using doubly linked nodes.

2.1 Constructor

* Initializes **start** and **last** pointers to **NULL** in the constructor.

2.2 **input** Method

* Takes a binary string as input and creates nodes for each bit.
* Adds nodes to the end of the list.

2.3 **one\_complement** Method

* Performs a one's complement operation on the binary list by flipping each bit.

2.4 **two\_complement** Method

* Performs a two's complement operation on the binary list by first computing the one's complement and then adding 1.

2.5 **print** Method

* Prints the binary bits in the list.

2.6 **add\_bit\_at\_begin** Method

* Adds a binary bit at the beginning of the list.

2.7 Overloaded **+** Operator

* Overloads the **+** operator to perform binary addition of two lists.
* Implements a bitwise addition with carry.

**3. main Function**

* Creates instances of the **List** class (**l**, **l1**, **l2**).
* Accepts a binary number as input.
* Provides a menu for operations like one's complement, two's complement, and binary addition.
* Implements a loop to perform operations based on user input.

**Note:**

* The program uses a linked list to represent binary numbers and performs various operations on them.
* The **+** operator is overloaded to add two binary numbers.
* The **one\_complement** and **two\_complement** methods modify the list in-place.
* The program offers a menu-driven approach for users to perform different operations and repeats until the user chooses to exit.

Algorithm:Top of Form

1. \*Node Class:\*

- Public attributes: bit (boolean), next\_node (pointer to the next node), prev\_node (pointer to the previous node).

- Constructor initializes bit to 0, next\_node and prev\_node to NULL.

2. \*List Class:\*

- Private attributes: start (pointer to the first node), last (pointer to the last node).

- Constructor initializes start and last to NULL.

- \*\*input Function:\*\*

- Takes a binary string as input from the user.

- Creates nodes for each binary digit and appends them to the linked list.

- \*\*one\_complement Function:\*\*

- Inverts each bit in the linked list.

- \*\*two\_complement Function:\*\*

- Calls one\_complement to get the one's complement.

- Adds 1 to the result to get the two's complement.

- \*\*print Function:\*\*

- Prints the binary number represented by the linked list.

- \*\*add\_bit\_at\_begin Function:\*\*

- Adds a new node with the specified bit at the beginning of the linked list.

- \*\*operator+ Overloaded Function:\*\*

- Adds two binary numbers represented by linked lists.

- Uses bitwise XOR to calculate the sum bit.

- Handles carry propagation.

3. \*\*main Function:\*\*

- Creates instances of the List class: l, l1, and l2.

- Enters a loop to perform operations until the user chooses to exit.

- Displays a menu to the user and takes input for the desired operation.

- Performs the selected operation and prints the result.

- Asks the user if they want to try again.

4. \*Sample Operations:\*

- For option 1, it performs one's complement twice to revert the changes.

- For option 2, it performs two's complement twice to revert the changes.

- For option 3, it takes another binary number as input and performs addition.