

//CODE:

```
#include <iostream>

#include <string>

using namespace std;

class Node
{
public:
    int bit;
    Node* next;
    Node* prev;
    Node(int b) : bit(b), next(nullptr), prev(nullptr) {}
};

class DoublyLinkedList
{
public:
    Node* head;
    Node* tail;
    DoublyLinkedList() : head(nullptr), tail(nullptr) {}
    void append(int bit) {
        Node* newNode = new Node(bit);
        if (!head) {
            head = newNode;
            tail = newNode;
        }
        else {
            tail->next = newNode;
            newNode->prev = tail;
            tail = newNode;
        }
    }
};
```

```

}

void display() {
    Node* temp = head;
    while (temp) {
        cout << temp->bit;
        temp = temp->next;
    }
    cout << endl;
}

void onesComplement() {
    Node* temp = head;
    while (temp) {
        temp->bit = (temp->bit == 0) ? 1 : 0;
        temp = temp->next;
    }
}

void twosComplement() {
    onesComplement();
    addBinary(1);
}

void addBinary(int bit) {
    Node* temp = tail;
    int carry = bit;
    while (temp) {
        int sum = temp->bit + carry;
        temp->bit = sum % 2;
        carry = sum / 2;
        if (carry == 0) break;
        temp = temp->prev;
    }
}

```

```

    }
    if (carry > 0) {
        append(carry);
    }
}

void addBinary(DoublyLinkedList& other) {
    Node* temp1 = tail;
    Node* temp2 = other.tail;
    int carry = 0;
    while (temp1 || temp2 || carry) {
        int sum = carry;
        if (temp1) {
            sum += temp1->bit;
            temp1 = temp1->prev;
        }
        if (temp2) {
            sum += temp2->bit;
            temp2 = temp2->prev;
        }
        append(sum % 2);
        carry = sum / 2;
    }
}

};

```

```

int main() {
    DoublyLinkedList binaryNumber1;
    DoublyLinkedList binaryNumber2;
    int choice;

```

```

do {
    cout << "\nMenu:\n";
    cout << "1. Input first binary number\n";
    cout << "2. Input second binary number\n";
    cout << "3. Compute 1's complement of the first binary number\n";
    cout << "4. Compute 2's complement of the first binary number\n";
    cout << "5. Add the two binary numbers\n";
    cout << "6. Exit\n";
    cout << "Enter your choice: ";
    cin >> choice;

    switch (choice) {
        case 1: {
            string input1;
            cout << "Enter first binary number: ";
            cin >> input1;
            for (char c : input1) {
                binaryNumber1.append(c - '0');
            }
            break;
        }
        case 2: {
            string input2;
            cout << "Enter second binary number: ";
            cin >> input2;
            for (char c : input2) {
                binaryNumber2.append(c - '0');
            }
        }
    }
}

```

```

        break;
    }
    case 3:
        cout << "1's Complement of the first binary number: ";
        binaryNumber1.onesComplement();
        binaryNumber1.display();
        break;
    case 4:
        cout << "2's Complement of the first binary number: ";
        binaryNumber1.twosComplement();
        binaryNumber1.display();
        break;
    case 5: {
        DoublyLinkedList sum;
        sum.addBinary(binaryNumber1);
        sum.addBinary(binaryNumber2);
        cout << "Sum of both binary numbers: ";
        sum.display();
        break;
    }
    case 6:
        cout << "Exiting...\n";
        break;
    default:
        cout << "Invalid choice, please try again.\n";
    }
} while (choice != 6);
return 0;
}

```

//OUTPUT:

Menu:

1. Input first binary number
2. Input second binary number
3. Compute 1's complement of the first binary number
4. Compute 2's complement of the first binary number
5. Add the two binary numbers
6. Exit

Enter your choice: 1

Enter first binary number: 10011

Menu:

1. Input first binary number
2. Input second binary number
3. Compute 1's complement of the first binary number
4. Compute 2's complement of the first binary number
5. Add the two binary numbers
6. Exit

Enter your choice: 2

Enter second binary number: 11001

Menu:

1. Input first binary number
2. Input second binary number
3. Compute 1's complement of the first binary number
4. Compute 2's complement of the first binary number
5. Add the two binary numbers
6. Exit

Enter your choice: 3

1's Complement of the first binary number: 01100

Menu:

1. Input first binary number
2. Input second binary number
3. Compute 1's complement of the first binary number
4. Compute 2's complement of the first binary number
5. Add the two binary numbers
6. Exit

Enter your choice: 4

2's Complement of the first binary number: 10100

Menu:

1. Input first binary number
2. Input second binary number
3. Compute 1's complement of the first binary number
4. Compute 2's complement of the first binary number
5. Add the two binary numbers
6. Exit

Enter your choice: 5

Sum of both binary numbers: 0010101111

Menu:

1. Input first binary number
2. Input second binary number
3. Compute 1's complement of the first binary number
4. Compute 2's complement of the first binary number
5. Add the two binary numbers
6. Exit

Enter your choice: 6

Exiting...