Assignment 4

Name : Om D. Tambat

Enrollment No. : BT22CSH042

Q1.

Code:

```
#include <iostream>
#include <string>

struct Node {
   int data;
   Node* next;
   Node* prev;
};

class DoublyLinkedList {
private:
   Node* head;
   Node* tail;

public:
   DoublyLinkedList() {
      head = tail = NULL; // Use NULL instead of nullptr
   }

   void insertAtEnd(int digit) {
      Node* newNode = new Node;
```

```
        newNode->data = digit;
        newNode->next = NULL; // Use NULL instead of nullptr
        newNode->prev = tail;

        if (!head) {
            head = tail = newNode;
        } else {
            tail->next = newNode;
            tail = newNode;
        }
    }

    void displayReverse() {
        Node* current = tail;
        while (current) {
            std::cout << current->data;
            current = current->prev;
        }
        std::cout << std::endl;
    }

    static DoublyLinkedList addNumbers(const DoublyLinkedList& num1, const
DoublyLinkedList& num2) {
        DoublyLinkedList result;
        int carry = 0;

        Node* n1 = num1.tail;
```

```
        Node* n2 = num2.tail;

        while (n1 || n2) {
            int sum = carry + (n1 ? n1->data : 0) + (n2 ? n2->data : 0);
            carry = sum / 10;
            result.insertAtEnd(sum % 10);

            if (n1) n1 = n1->prev;
            if (n2) n2 = n2->prev;
        }

        if (carry) {
            result.insertAtEnd(carry);
        }

        return result;
    }
};

int main() {
    std::cout << "Enter the first number: ";
    std::string num1Str;
    std::cin >> num1Str;

    std::cout << "Enter the second number: ";
    std::string num2Str;
```

```
    std::cin >> num2Str;

    DoublyLinkedList num1, num2;

    for (char digit : num1Str) {
        num1.insertAtEnd(digit - '0');
    }

    for (char digit : num2Str) {
        num2.insertAtEnd(digit - '0');
    }

    std::cout << "Number 1: ";
    num1.displayReverse();
    std::cout << "Number 2: ";
    num2.displayReverse();

    DoublyLinkedList sum = DoublyLinkedList::addNumbers(num1, num2);

    std::cout << "Sum: ";
    sum.displayReverse();

    return 0;
}
```

Output :


Output
Clear
/tmp/ejojpGKKBm.o
Enter the first number: 5432
Enter the second number: 6789
Number 1: 2345
Number 2: 9876
Sum: 12221