BT22CSH001 SAUMYA KUMAR

QUESTION 1-

[*] Untitled1

```cpp
#include <bits/stdc++.h>
using namespace std;

struct MatrixNode {
    int value;
    int column;
    int row;
    MatrixNode* next;
};

void insert(MatrixNode* nodePtr, int val, int r, int c) {
    while (nodePtr->next) {
        nodePtr = nodePtr->next;
    }
    MatrixNode* newNode = new MatrixNode;
    nodePtr->next = newNode;
    newNode->value = val;
    newNode->column = c;
    newNode->row = r;
    newNode->next = nullptr;
}

void display(MatrixNode* nodePtr) {
    while (nodePtr) {
        cout << nodePtr->value << " " << nodePtr->row << " " << nodePtr->column << endl;
        nodePtr = nodePtr->next;
    }
}

void transpose(MatrixNode* nodePtr) {
    while (nodePtr) {
        swap(nodePtr->row, nodePtr->column);
        nodePtr = nodePtr->next;
    }
}

void displayMatrix(MatrixNode* nodePtr, int numRows, int numCols) {
    cout << endl;
    int matrix[numRows][numCols];
    for (int i = 0; i < numRows; i++) {
        for (int j = 0; j < numCols; j++) {
            matrix[i][j] = 0;
        }
    }
    while (nodePtr) {
        matrix[nodePtr->row - 1][nodePtr->column - 1] = nodePtr->value;
        nodePtr = nodePtr->next;
    }
    for (int i = 0; i < numRows; i++) {
```

[*] Untitled1

```
    }
}

void displayMatrix(MatrixNode* nodePtr, int numRows, int numCols) {
    cout << endl;
    int matrix[numRows][numCols];
    for (int i = 0; i < numRows; i++) {
        for (int j = 0; j < numCols; j++) {
            matrix[i][j] = 0;
        }
    }
    while (nodePtr) {
        matrix[nodePtr->row - 1][nodePtr->column - 1] = nodePtr->value;
        nodePtr = nodePtr->next;
    }
    for (int i = 0; i < numRows; i++) {
        for (int j = 0; j < numCols; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << "\n";
    }
}

int main() {
    MatrixNode* matrixList = new MatrixNode;
    matrixList->next = nullptr;

    int numElements;
    cout << "Enter the number of rows and columns in a matrix:\n";
    int numRows, numCols;
    cin >> numRows >> numCols;
    cout << "Enter the number of non-zero elements:\n";
    cin >> numElements;

    for (int i = 0; i < numElements; i++) {
        int val, r, c;
        cout << "Enter element, row, and column:\n";
        cin >> val >> r >> c;
        insert(matrixList, val, r, c);
    }

    matrixList = matrixList->next;
    displayMatrix(matrixList, numRows, numCols);
    transpose(matrixList);
    displayMatrix(matrixList, numRows, numCols);

    return 0;
}
```

```
Enter the number of rows and columns in a matrix
3
4
Enter the number of non-zero elements
5
Enter element, row, and column
2 2  2
Enter element, row, and column
4 1 2
Enter element, row, and column
3 1 3
Enter element, row, and column
2 3 1
Enter element, row, and column
1 1 1

1 4 3 0
0 2 0 0
2 0 0 0
```

QUESTION-2

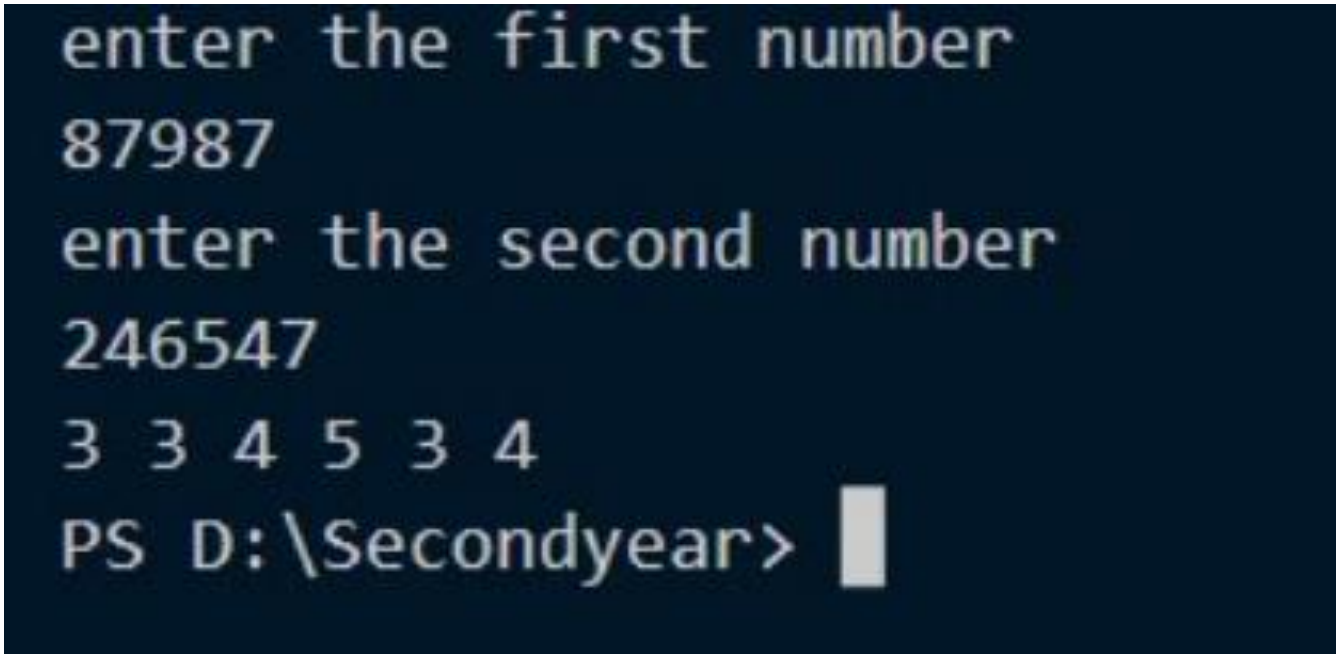
[*] Untitled1

```cpp
#include <bits/stdc++.h>
using namespace std;

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

Node* createNode(Node* p, int n) {
    Node* t;
    p->data = n % 10;
    p->next = NULL;
    n = n / 10;
    while (n) {
        t = new Node;
        t->next = p;
        t->data = n % 10;
        p = t;
        n = n / 10;
    }
    return p;
}

Node* reverseNode(Node* p) {
    Node* q = NULL;
    Node* r = NULL;
    while (p != NULL) {
        r = q;
        q = p;
        p = p->next;
        q->next = r;
    }
    return q;
}

void displayNode(Node* p) {
    while (p) {
        cout << p->data << " ";
        p = p->next;
    }
    cout << endl;
}

void addNumbers(Node* p, Node* q) {
    Node* result = NULL;
    int carry = 0;
    int sum = 0;
    while (p || q || carry) {
        sum = 0;
```

er Resources Compile Log Debug Find Results

[*] Untitled1

```
46      int carry = 0;
47      int sum = 0;
48      while (p || q || carry) {
49          sum = 0;
50          if (p)
51              sum += p->data;
52          if (q)
53              sum += q->data;
54          if (carry)
55              sum += carry;
56          Node* t = new Node;
57          t->data = sum % 10;
58          carry = sum / 10;
59          if (!result) {
60              result = t;
61              t->next = NULL;
62              if (p)
63                  p = p->next;
64              if (q)
65                  q = q->next;
66              continue;
67          }
68          t->next = result;
69          result = t;
70          if (p)
71              p = p->next;
72          if (q)
73              q = q->next;
74      }
75      displayNode(result);
76  }
77
78  int main() {
79      Node* first = new Node;
80      Node* second = new Node;
81      int n;
82      cout << "Enter the first number: ";
83      cin >> n;
84      int n2;
85      cout << "Enter the second number: ";
86      cin >> n2;
87      second = createNode(second, n2);
88      first = createNode(first, n);
89      Node* revFirst = reverseNode(first);
90      Node* revSecond = reverseNode(second);
91      addNumbers(revFirst, revSecond);
92      return 0;
93  }
94
```


enter the first number
87987
enter the second number
246547
3 3 4 5 3 4
PS D:\Secondyear>