**Assignment 3**

**GitHub Link:**

[https://github.com/sufiyanjunaidi13/Advance\_Algorithm\_Assignment-3 (github.com)](https://github.com/sufiyanjunaidi13/Advance_Algorithm_Assignment-3)

**#Program to Implement a LinkedList Implementation for Sparse Matrices**

#include <iostream>

using namespace std;

// Node class to represent link list

class Node {

public:

int row;

int col;

int data;

Node\* next;

};

// Function to create new node

Node\* createNewNode(int row\_index, int col\_index, int x) {

Node\* newNode = new Node();

newNode->row = row\_index;

newNode->col = col\_index;

newNode->data = x;

newNode->next = NULL;

return newNode;

}

// Function to insert node in linked list

void insertNode(Node\*\* head, int row\_index, int col\_index, int x) {

Node\* temp = \*head;

// If link list is empty then

// create first node and assign value.

if (temp == NULL) {

\*head = createNewNode(row\_index, col\_index, x);

}

// If link list is already created

// then append newly created node

else {

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = createNewNode(row\_index, col\_index, x);

}

}

// Function to print contents of linked list

void printList(Node\* head) {

Node\* ptr = head;

cout << "row\_position:";

while (ptr != NULL) {

cout << ptr->row << " ";

ptr = ptr->next;

}

cout << endl;

cout << "column\_position:";

ptr = head;

while (ptr != NULL) {

cout << ptr->col << " ";

ptr = ptr->next;

}

cout << endl;

cout << "Value:";

ptr = head;

while (ptr != NULL) {

cout << ptr->data << " ";

ptr = ptr->next;

}

}

// Driver Code

int main() {

// 4x5 sparse matrix

int sparseMatrix[4][5] = { { 0, 0, 3, 0, 4 },

{ 0, 0, 5, 7, 0 },

{ 0, 0, 0, 0, 0 },

{ 0, 2, 6, 0, 0 } };

// Creating head/first node of list as NULL

Node\* head = NULL;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 5; j++) {

// Pass only those values which

// are non-zero

if (sparseMatrix[i][j] != 0) {

insertNode(&head, i, j, sparseMatrix[i][j]);

}

}

}

printList(head);

return 0;

}

