Roll NO:27

NAME : Mali Anjali Prakash

DIV: FYMCA-A

Assignment no -1

#include <stdio.h>

#define MAX\_ELEMENTS 100

// Structure to store a non-zero element of the matrix

typedef struct {

    int row;

    int col;

    int value;

} SparseElement;

// Function to perform sparse matrix transpose

void fastTranspose(SparseElement matrix[], int \*numElements, SparseElement transposed[], int numRows, int numCols) {

    int count[MAX\_ELEMENTS] = {0}; // Count the number of elements in each column

    int pos[MAX\_ELEMENTS]; // Position of each element in the transposed matrix

    // Count non-zero elements in each column

    for (int i = 0; i < \*numElements; i++) {

        count[matrix[i].col]++;

    }

    // Calculate position of each element in transposed matrix

    pos[0] = 0;

    for (int i = 1; i < numCols; i++) {

        pos[i] = pos[i - 1] + count[i - 1];

    }

    // Fill the transposed matrix using the calculated positions

    for (int i = 0; i < \*numElements; i++) {

        int j = matrix[i].col;

        transposed[pos[j]].row = matrix[i].col;

        transposed[pos[j]].col = matrix[i].row;

        transposed[pos[j]].value = matrix[i].value;

        pos[j]++;

    }

}

// Function to display a sparse matrix

void displaySparseMatrix(SparseElement matrix[], int numElements) {

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

        printf("Row: %d, Col: %d, Value: %d\n", matrix[i].row, matrix[i].col, matrix[i].value);

    }

}

int main() {

    int numRows, numCols, numElements;

    SparseElement matrix[MAX\_ELEMENTS];

    SparseElement transposed[MAX\_ELEMENTS];

    // Input matrix dimensions and non-zero elements

    printf("Enter the number of rows and columns of the matrix: ");

    scanf("%d %d", &numRows, &numCols);

    printf("Enter the number of non-zero elements: ");

    scanf("%d", &numElements);

    printf("Enter the non-zero elements (row, column, value):\n");

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

        scanf("%d %d %d", &matrix[i].row, &matrix[i].col, &matrix[i].value);

    }

    printf("\nOriginal Sparse Matrix:\n");

    displaySparseMatrix(matrix, numElements);

    // Perform fast transpose

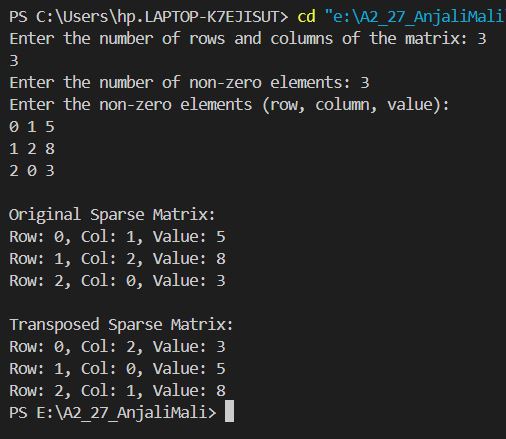
    fastTranspose(matrix, &numElements, transposed, numRows, numCols);

    printf("\nTransposed Sparse Matrix:\n");

    displaySparseMatrix(transposed, numElements);

    return 0;

}



Output: