#include <iostream>

using namespace std;

class SparseMatrix {

private:

int numRows, numCols, numNonZero;

int\*\* matrix;

public:

SparseMatrix(int rows, int cols) {

numRows = rows;

numCols = cols;

numNonZero = 0;

// Allocate memory for the 2D array

matrix = new int\*[rows];

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

matrix[i] = new int[cols];

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

matrix[i][j] = 0; // Initialize all elements to 0

}

}

}

void insert(int row, int col, int value) {

if (row < 0 || row >= numRows || col < 0 || col >= numCols) {

cout << "Invalid row or column index." << endl;

return;

}

matrix[row][col] = value;

if (value != 0) {

numNonZero++;

}

}

void display() {

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

int getNumRows() {

return numRows;

}

int getNumCols() {

return numCols;

}

int getNumNonZero() {

return numNonZero;

}

};

int main() {

int rows, cols;

cout << "Enter the number of rows and columns: ";

cin >> rows >> cols;

SparseMatrix sparseMatrix(rows, cols);

int row, col, value;

cout << "Enter the number of non-zero elements: ";

cin >> value;

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

cout << "Enter the row, column, and value for non-zero element " << i + 1 << ": ";

cin >> row >> col >> value;

sparseMatrix.insert(row, col, value);

}

cout << "Sparse Matrix:" << endl;

sparseMatrix.display();

cout << "Number of rows: " << sparseMatrix.getNumRows() << endl;

cout << "Number of columns: " << sparseMatrix.getNumCols() << endl;

cout << "Number of non-zero elements: " << sparseMatrix.getNumNonZero() << endl;