Lab Assignment: Fast Transpose of a Sparse Matrix (Tuple Representation)

Objective

Write a C program to find the **fast transpose** of a sparse matrix represented in **tuple form**, using the **counting sort technique**.

Sparse Matrix Tuple Representation

A sparse matrix is represented as a list of tuples: (row, col, value). The first entry is the header: (m, n, t), where m = number of rows, n = number of columns, t = number of non-zero elements.

Matrix	Tuple Form
[[0 0 3], [4 0 0], [0 5 6]]	(3,3,4), (0,2,3), (1,0,4), (2,1,5), (2,2,6)

Algorithm (Fast Transpose using Counting Sort)

1. Input the sparse matrix in tuple form.
2. Initialize an array count[col] to store the number of non-zero elements in each column.
3. Compute starting positions of each column in the transposed matrix using cumulative sums in count.
4. For each tuple (row, col, val) in the original matrix (ignoring the header): - Place it at the calculated position in the result array as (col, row, val). - Increment the position counter for that column.
5. Output the transposed sparse matrix in tuple form.

Function Signature (C)

```
void fastTranspose(int original[][3], int transposed[][3]);
```

Example

Input (Tuple Form):

```
4 4 5
0 1 10
0 3 12
1 2 5
2 3 15
3 0 6
```

Output (Fast Transposed Tuple Form):

```
4 4 5
0 3 6
1 0 10
2 1 5
3 0 12
3 2 15
```