Object Oriented Programming
Spring 2022
Assignment # 2
Submission date: Wednesday March 16, 2022

You are required to implement a program using C++ double Pointers to perform operation on sparse matrices. A **sparse matrix** is a matrix populated primarily with zeros.   Below is an example of 5*5 sparse matrix.

5	0	0	1	0
2	0	15	9	13
0	3	57	0	0
0	0	10	3	0
0	3	0	0	0

In order to save space, instead of having a whole nxm matrix in a two dimensional array you will use double pointers and allocate memory dynamically as per requirement. This means if a row in a matrix has only two non-zero entries, then memory to store two elements must be allocated dynamically.

Your program should perform following functions on these matrices.
1. Output matrix
2. Input matrix
3. Compute Transpose of the given matrix
4. Compute the resultant of addition of the two matrices
5. Compute the resultant of multiplication of two matrices

Input:
The input function must take input from a text file. The first line would give you the dimensions of the matrix. And rest of the line gives the information about the data in each row. For example

5	5									
5	0	3	1	5	2	11	3	13	4	20
2	0	15	2	13						
1	3	57								
3	1	10	3	24	4	67				
2	3	69	4	18						

This input tells that it is a 5*5 matrix. First element of each row tells the number of nonzero columns in that row. In above example the last row contains just 2 columns at third column the data value is 69 and in fourth column the data value is 18. For the rest of the columns, you can assume zeros.

Output:
Output of a matrix must include zeros e.g the above matrix would be

3	5	11	13	20

15	0	13	0	0
0	0	0	57	0
0	10	0	24	67
0	0	0	69	18

Transpose:
The transpose function must transpose the sparse matrix. This means the original matrix must be changed.

Addition:
The addition function takes two sparse matrices as parameter and compute the sum if both matrices are compatible for addition. Two matrices are compatible if their rows and columns are same. This function must return the resultant matrix if addition is possible and nullptr otherwise.

Multiplication:
The multiplication function takes two sparse matrices as parameter and compute the product if both matrices are compatible for multiplication. Two matrices are compatible if their columns of first matrix are same as rows of the other matrix. This function must return the resultant matrix if multiplication is possible and nullptr otherwise.

Note:
You can use white space as delimiter for you input.
You can not use anything other than double pointers for this assignment.
The resultant matrices must also be stored as sparse matrix and you are not allowed to convert them into ordinary matrices for any operation
Proper dynamic memory allocation and deallocation are part of this work. There must not be memory leaks and dangling pointers.