Documentation:

def transpose\_of\_csr(matrix\_in\_csr):

arg: Takes a matrix in CSR format and transpose it by switching the rows and colums

returns: a new matrix in CST format

def transpose(matrix):

arg: takes a matrix and transpose it by converting the columns into rows

def multiply\_two\_matrices(first\_matrix, second\_matrix):

arg: takes two matrices and multiply them

returns a new matrix

def multiply\_matrix\_times\_vector(a\_matrix, vector):

arg: takes a matrix and a vector

return a new matrix

def read\_from\_file(name\_of\_file):

arg: takes the name of the file

return a list of lists where each list is a single line in the file

def write\_to\_file(name\_of\_file, a\_list):

arg: takes the name of the file and a list of lists

writes each list in a single line of the file

def multi\_two\_csr(csr\_1, csr\_2):

arg: takes two matrices in csr format

and multiply them

the row of the first csr and the column of the second csr are where it stores the result

of the multiplication of the column of the csr\_1 time the row of csr\_2.

The result is stored in a dictionary. The key is the row and column indices and the value is

The result. Each time we try to store a result in a dictionary, we check if that key exists already.

If so, we add the existing value plus a new one.

And the end, we return the multiplication in csr format