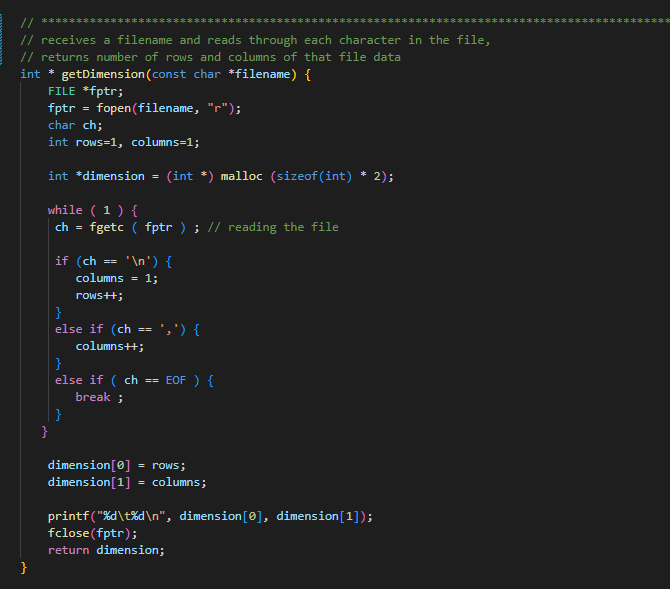
# Task 1

## Reading data from file

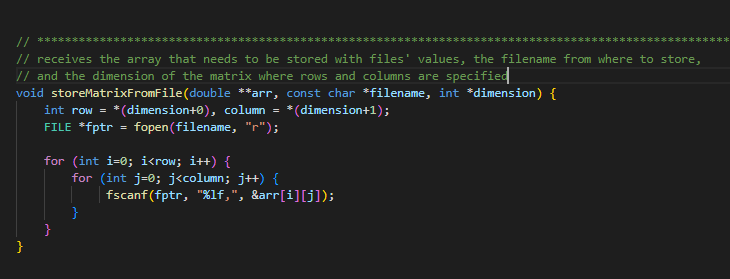
While reading data from given sample text files, there are two process of reading matrix data.

1. **For getting dimension of matrix**



This function receives a filename which has all the matrix data stored in it. It reads through each character in the matrix text file and determines total number of rows and columns. The final count of rows and columns are sent back in an array named dimension where first index stores row count and second index stores column value.

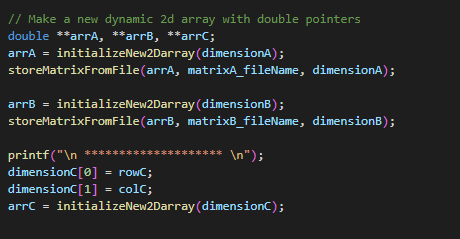
1. **For reading data and storing it in the double ‘double array’**

****

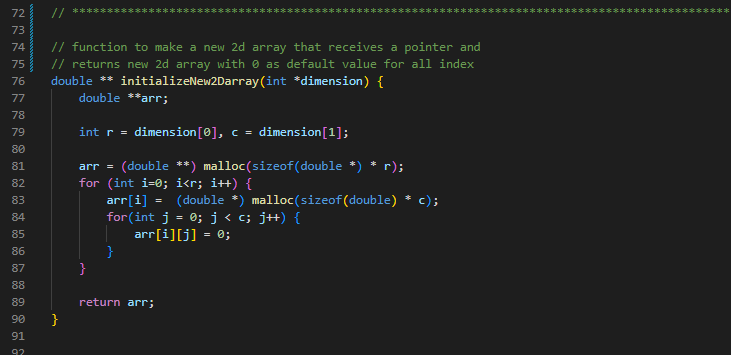
This function stores data from matrix file to double double array. It receives the matrix name from where the data is to be copied, the double pointers double array where it needs to get stored and the dimension of that matrix which we got from the function above. The initialization and declaration of that double pointers double matrix is explained below.

## Using dynamic memory allocation for all matrices

I declared a double pointers array for each matrix and stored 0 as default value to each index of that 2-d array. This how the functions are called, after declaration. Where initializeNew2Darray() is described below and storeMatrixFromFile() is explained above.



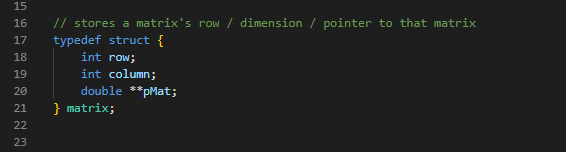
1. **Use of double pointers `double array`**



In this function, it receives dimension of the matrix and returns a new double pointer double matrix with rows and columns divides as chunks. It creates a double pointer dynamic array with size of total rows \* (double \*) double pointer. Then for each row, it declares value of dynamic array, with size of double \* total columns.

For a column there are chunks of double pointer rows and in that row, there exists chunks of double datatype value. All the values are of datatype double. Finally, it returns the double pointer to double array.

1. **Use of struct to manage matrix data**

****

This structure holds the row data, column data and the actual double pointer double array of a matrix.

In this function, it receives a matrix which is of type struct. It contains fields like row, column and the actual pointer to the 2-d array.

## Matrix multiplication thread function

## Multithreading usage for equal computation

## Correct output display and matrix storage

# Task 2

# Task 3

# Task 4