**Problem statement:**

Parallelize the multiplication of 3 matrices. Use a configurable number of threads to do one matrix multiplication. Then, use another set of threads to do the second multiplication. The threads in the second set should start as soon as they start having data from the first multiplication result.

**Solution:**

In order to perform the multiplication of 3 matrices in parallel I choose the following approach:

* At the beginning I initialized two thread pull with a given number of threads
* Also I initialized a list of tasks for both thread pull
  + So the first list will contain the pair<lineIndex,columnIndex> which represents the line and the column which will be multiply in order to obtain the result corresponding to [lineIndex][columnIndex] resProd1, the lineIndex and columnIndex will be in the range (nr of line from A, nr of column for B)
  + The second line will contain also the pair of the form <lineIndex,columnIndex> but now this will coresponding whit the result from the third multiplication, lineIndex and columnIndex will be in the range (nr of line from resProd1, nr of column for C)
* I declare two global mutex mutexx1, mutexx2 which will block the lists of tasks in order to be able to take a tasks from this in a parralel way.
* I have also declare a list of integer which will have the dimension equal with the number of rows from the result. In this list I will mark every time when I perform a task over lineIndex. In this way I will know when I can start to do the third multiplication
* mutexxM list is designed in order to provide a safe way to access the element from the list previous declare.

The flow of the program will be as follow:

1. I am a thread from the first thread pool -> I will take a task from the corresponding list:
   1. If I can( the list is not empty) I will take the lock over the corresponding line and after that do the multiplication and release the lock
   2. If not->just stop, I done my work
2. I am a thread from the second thread pool -> I will take a task from the corresponding list:
   1. If I can( the list is not empty) I will take the lock over the corresponding line and

check if from this line all the elements were computed( checked if the value of countt[indexLine]=number of elements from column)

* If yes-> perform the corresponding multiplication
* If not-> release the lock over the current line and put the current task in the list of task( it is not ready to be perfom)
  1. If not->just stop, I done my work