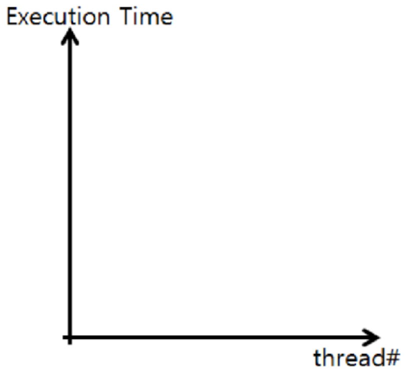
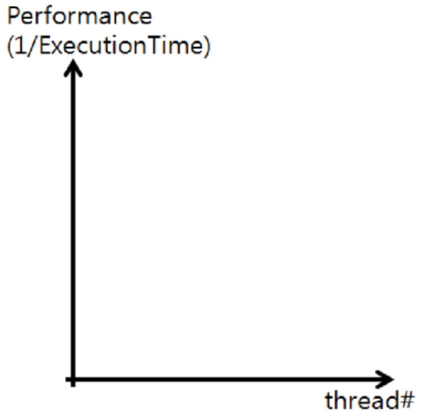
problem 2. (i) Given a JAVA source code for matrix multiplication (the source code MatmultD.java is available on our class webpage), modify the JAVA code to implement parallel matrix multiplication that uses multi-threades. You may use either a static or a dynamic load balancing approach. Your code also should print (1) the execution time of each thread, (2) execution time for the entire thread computation, and (3) sum of all elements in the resulting matrix. Use the matrix mat500.txt (available on our class webpage) as file input (standard input) for the performance evaluation. mat500.txt contains two matrices that will be used for multiplication.

command line execution example in cygwin terminal> java MatmultD 6 < mat500.txt
In eclipse, set the argument value and file input by using the menu [Run]->[Run Configurations]->{[Arguments], [Common -> Input File].

Here, 6 means the number of threads to use, < mat500.txt means the file that contains two matrices is given as standard input.

(ii) Write a document that reports the parallel performance of your code. The graph that shows the execution time when using 1, 2, 4, 6, 8, 10, 12, 14, 16 threads. Your document also should mention which CPU (dualcore? or quadcore?, clock speed) was used for executing your code.





	1	2	4	...	16
exec time					

	1	2	4	...	16
performace (1/exec time)					