HOMEWORK 4: PARALLEL PROGRAMMING WITH OPENMP (2)

DEADLINE: JANUARY 30

1. Parallelise the following algorithms using OpenMP:
   a. Gaussian elimination algorithm for solving a linear system of equations assuming that the pivot element cannot be zero: http://en.wikipedia.org/wiki/Gaussian_elimination;
   b. Dijkstra's algorithm for computing the shortest path between nodes in a graph stored as an matrix: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm;
   c. Sieve of Eratosthenes for finding all prime numbers to a given limit: https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes;
   d. Quick sort: https://en.wikipedia.org/wiki/Quicksort.

2. Choose for each algorithm one large problem size (i.e. array dimension), initialise the array with uniformly distributed random numbers;

3. Execute the parallel algorithm on the `stud1.itec.aau.at` parallel machine using the `Slurm` workload manager with 1, 2, 4 and 8 parallel threads;

4. Report the execution time, speedup and efficiency metrics in a simple PDF file.

Important requirements:

1. Measure only the core execution time of each algorithm without random number generation, array/matrix initialisation, and any I/O operations.
2. Declare the main array or matrix data structures as static global variables. Do not use dynamic memory allocation using `malloc`.
3. Implement the algorithms as short (few lines) and as simple as possible, focused on the core functionality (e.g. without safety checks, small optimisations, or redundant tests).