OpenMP Spring Bonus (Gives bonus points for the final Exam!)[1]
Please send codes + brief report + timings before 25.04.2021

1. Simple embarrassingly parallel 2D Matrix computation. Compute each (i, j) value of a "big" A matrix (es: 1000 x 1000 double value):

$$A(i, j) = 15 * \cos(i)*\sin(j) * \sqrt{2*i} * \pi * j^6$$

2. Given three (huge, like 10^8 integer values) random vectors A, B and C, compute (adopt padding, intelligent critical sections, reduction, etc):

- C= A + B.
- Maximum of C

3. Calculation of PI
   A. Standard method (numerical integration + critical section + padding)
   B. Reduction
   C. Monte Carlo

4. Find an element in a vector (Extra BONUS - WOW! be careful !). HINT: Use `cancel` clause…

5. **[OPTIONAL]** Implement a simple version of the Game of Life (no need of graphic output). Remember slides for implementation hints (main and support matrixes, etc).
   – http://en.wikipedia.org/wiki/Conway's_Game_of_Life
   – https://www.youtube.com/watch?v=My8AsV7bA94
   – http://pmav.eu/stuff/javascript-game-of-life-v3.1.1/

*****

After implementing the versions adopting OpenMP instructions (`pragmas`, `parallel for`, etc), take timings and speed-up by:
a) **varying the dimensions** of the considered data structures/point (depends on the problem). Start by chosing an appropriate problem dimension.
b) **varying the number** of threads .
c) changing worksharing construct scheduling (static, dynamic, etc) where applicable.

For Example (Sum Vec – Static Scheduling):
Timings (seconds)

Threads / Dimension	10000000	100000000	1000000000		
1					
2					
4					

Speed-up (Ts/Tp)

Threads / Dimension	10000000	100000000	1000000000		
1					
2					
4					

1 Write a brief one/two page report of what was done + spreadsheet file. Please use appropriate multi-core machines paying attention to the number of adopted threads! (i.e., max 8 for a 4+4 hyperthread i7)