CENG444 **Spring 2023**

PROGRAMMING ASSIGNMENT 3

Due date: 14.06.2023 23:00

You are required to implement a parallel image blurring algorithm. The algorithm will take each pixel value in the original image and convert it into a new value to be placed in the same location in another image (do not modify the original input).

The new value will be calculated as the weighted sum of its original value and the original value of the 24 pixels that surround it. That is, your implementation will use a 5 × 5 patch. When calculating an output pixel value at the (Row, Col) position, the patch is centered at the input pixel located at the (Row, Col) position. The 5 × 5 patch spans five rows (Row-2, Row-1, Row, Row+1, Row+2) and five columns (Col-2, Col-1, Col, Col+1, Col+2).

You will use the following 5x5 Gaussian blur kernel (approximation) as the weights of the neighborhood elements. That is, you will multiply the corresponding neighbors by the values in the kernel box, sum up all of them, then divide the result by 273 to get the output image pixel value.

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

You can apply any policy for the boundary values by specifying it in your code/report.

You are required to implement a parallel version of this computation on OpenMP.

You can work with gray-scale images in any format. Your program does not need to support all image file formats, but you need to demonstrate your work for some images that you are required to submit as part of your submission. We will also test your program with other images in the same format. You can use available source codes/library functions to perform file operations by providing the appropriate references.

Notes:

- You are required to run your programs with large image files and various number of threads (1-16), and make a performance analysis by comparing different versions.

- You are required to write a report (2 pages at most), which includes
  - Your computer's specification (i.e., the number of cores, the number of threads; typically that can be obtained by *lscpu* command in Linux platform),
  - Instructions for compiling and executing your programs,
  - Graphs, your observations about the performance of your implementation, how you interpret the results.

Submission: You are required to submit your **commented** source code, report and sample images to cloud-lms. Please create a compressed file including all source files and report; and name it as yourstudentnumber_ASS3.zip (e.g. If your student number is 202312345678, the file name must be 202312345678_ASS3.zip). You need to work individually, no group work is allowed.