**CSC 453 introduction to parallel processing**

**Homework #4**

**Due date: 8-11-2020 (before 5:00 PM)**

Write a parallel program using OpenMP that calculate the ***PI*** value. Use the same code in Mattson tutorials. Run the code using p= 1,2,3,4 threads and with steps=100000, 1000000, 10000000 print the results in a table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RUN# | p | step | T1 | Tp | Speedup | Efficiency |
| 1 | 1 | 100000 |  |  |  |  |
| 2 | 2 | 100000 |  |  |  |  |
| 3 | 3 | 100000 |  |  |  |  |
| 4 | 4 | 100000 |  |  |  |  |
| 5 | 1 | 1000000 |  |  |  |  |
| 6 | 2 | 1000000 |  |  |  |  |
| 7 | 3 | 1000000 |  |  |  |  |
| 8 | 4 | 1000000 |  |  |  |  |
| 9 | 1 | 10000000 |  |  |  |  |
| 10 | 2 | 10000000 |  |  |  |  |
| 11 | 3 | 10000000 |  |  |  |  |
| 12 | 4 | 10000000 |  |  |  |  |

Watch these videos to have an idea how to solve the problem:

<https://www.youtube.com/watch?v=FQ1k_YpyG_A&list=PLLX-Q6B8xqZ8n8bwjGdzBJ25X2utwnoEG&index=6>

<https://www.youtube.com/watch?v=OuzYICZUthM&list=PLLX-Q6B8xqZ8n8bwjGdzBJ25X2utwnoEG&index=7>

Submit your report that include the following:

1. **Cover page:** Align the text that contains your name & ID, Course name and code, "Homework #4" in the center.
2. **Introduction:**  What is the problem to be solved?
3. **Body:** The body of your report should contain the main points from your work. Provide information about the code and how it was run in parallel, what OpenMP statements used and why, so that the reader can further understand your parallel code. How you measured the time. How many runs you made to take the average? Include the sample output of your program for different cases with important findings like **serial and parallel run time, the speedup, and efficiency.**
4. **Conclusion:** End with a summary of the results and findings as in the table above. Was the parallel code useful to enhance the execution time? If you have chance in the future what changes you may do for your code? Why?
5. **Appendices:** That include your coding and results
   1. Appendix A: The parallel code.
   2. Appendix B: The sample output.
   3. Appendix C: Table of the results of all runs.