Assignment 1: Report

Name: **<your-name-here>**

ID: **<your-student-id-here>**

Based on your solution and understanding of the assignment and lecture slides, fill out the questions below:

1. How are master and worker threads spawned? Briefly explain if they work together, how they synchronize, and what C++ features did you use to make this work. How many CPU cores are available on an Alma cluster node? How many threads can you use, and how many are used for the performance measurements?
2. How is the work distributed among the threads? Did you use static or dynamic work distribution? Is one performing better than the other and why?
3. What accesses needed to be protected with locks to ensure safe accesses?
4. In the version where you needed to use atomic variables, which variables needed to be used as atomic variables? Are there any variables where atomic types were not suitable?
5. Briefly explain how you implemented the SafeQ so that multiple threads can access it safely? Which accesses needed to be protected for this to work and why? In which methods, and why not in others? How did you make threads wait for more items?
6. Where do threads need to synchronize in the code so that results are correct?
7. Are there any bottlenecks/performance issues in your code? Which synchronization parts/mechanisms are causing the most overheads and why? What was the best-performing version? Did you overcome the performance issues and how? Is there any relation to the memory and caches?
8. Include a speedup graph (as shown in the slides) showing the speedup on ALMA for all provided versions. Use horizontal axis for the number of threads (1-32) and the vertical axis for speedup (not the execution time!).
9. Include a table with execution times and speedup of the parallel code. Speedup is measured compared to the sequential version (~10 seconds on ALMA nodes). If you have these in an Excel sheet, you can just attach it with your submission.

You write here any additional content and analysis that does not fit into above categories.