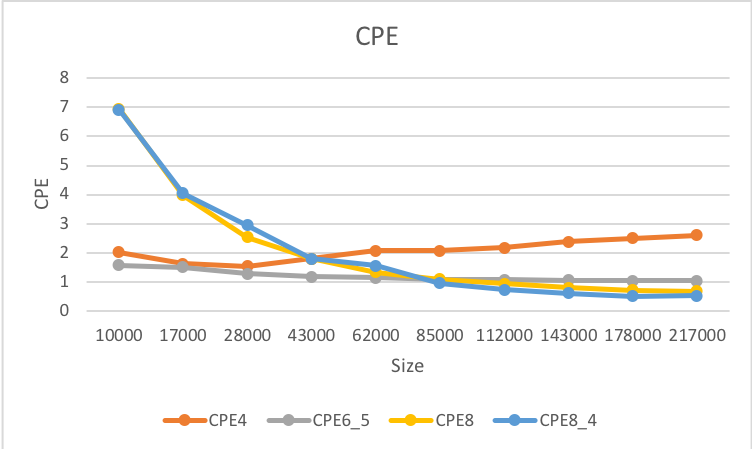
EC527 Assignment 3 Chen-Yu Chang U93093024

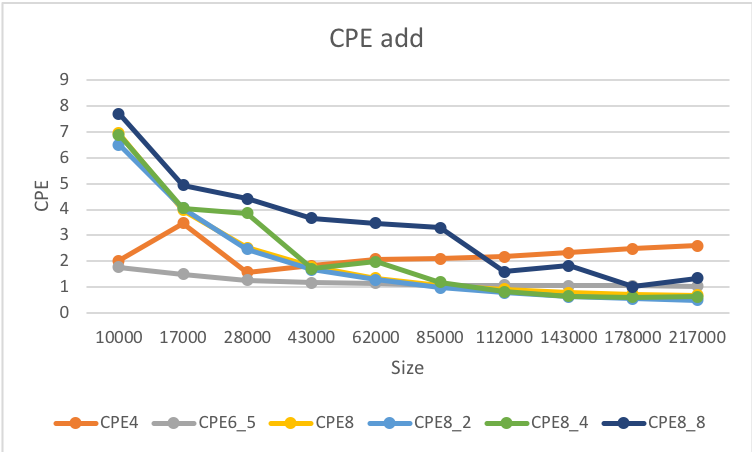
Part 1.

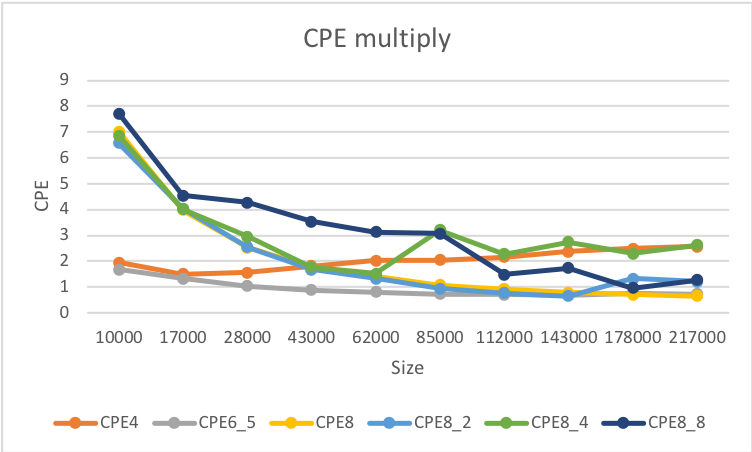
1a.



From the graph above, we can see that the fastest function is eventually combine8\_4. Comparing scalars and vectors, vector parallelization is definitely the more efficient way to work with since in a single cycle, there are more elements operating at a time.

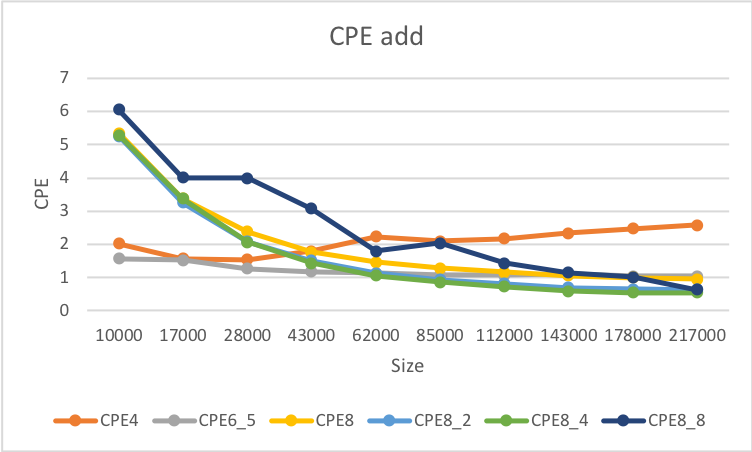
1b.

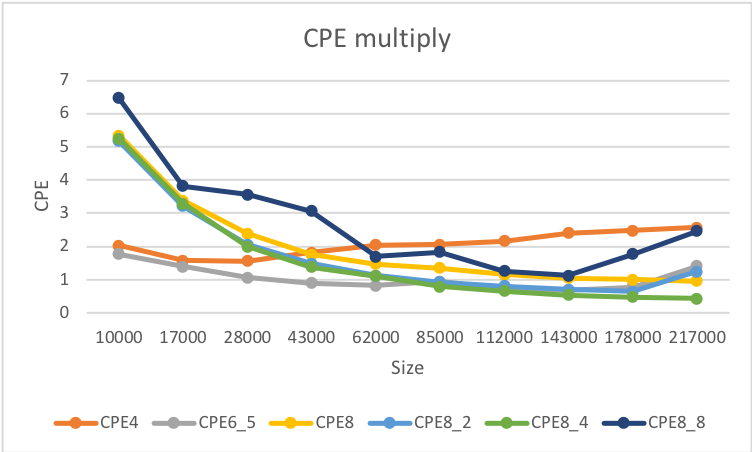




From the two graphs above of addition and multiplication, we can conclude, for addition, that combine8\_2 and combine8\_4 have the best performance since parallelization increases the efficiency. For multiplication, combine8 and combine6\_5 tend to perform the best. Therefore, we know that unrolling more times does help, but more than a certain number, it will not be helpful because CPU has only 5 stages.

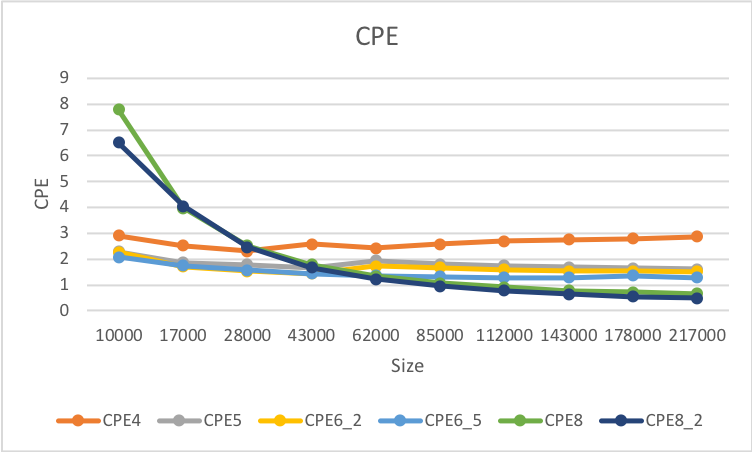
1c.





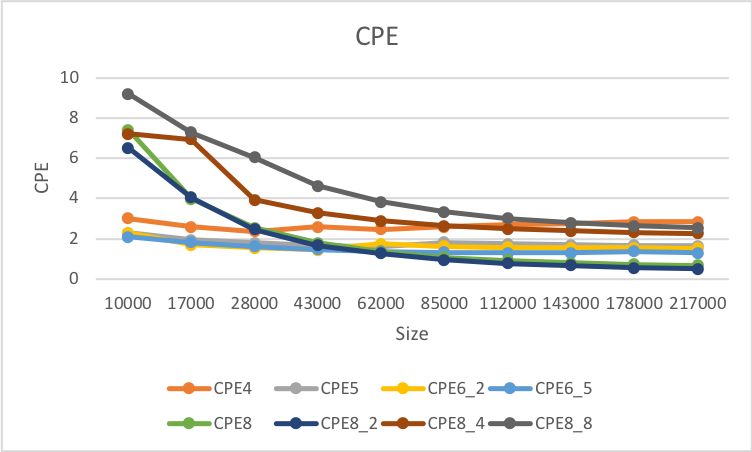
From the graphs above, overall, the combine8\_8 for double works better than for floats.

1d.



In the case of dot, the most stably efficient is having an unroll factor of 2 and 5 of scalar ones. However, when the size of the vector becomes bigger, the efficiency also improved due to the use of parallelization.

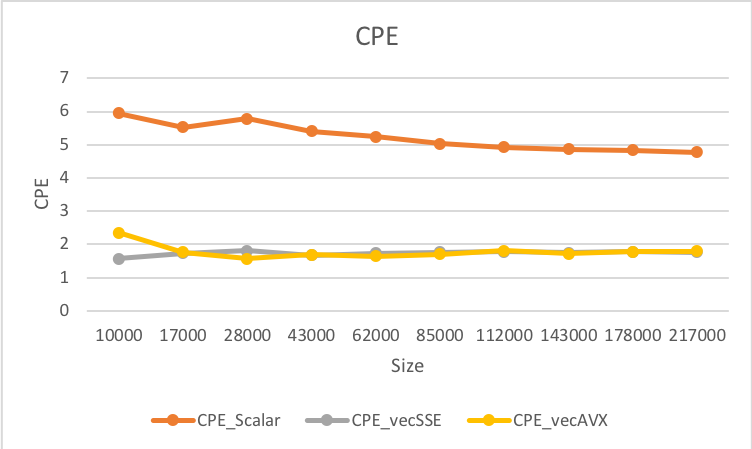
1e.



In this case, we see that even though we increase the unrolling factor, the CPE does not decrease. Therefore, having a high unrolling factor does not mean the higher efficiency of the operation. Still, combine8 and combine8\_2 tend to work better.

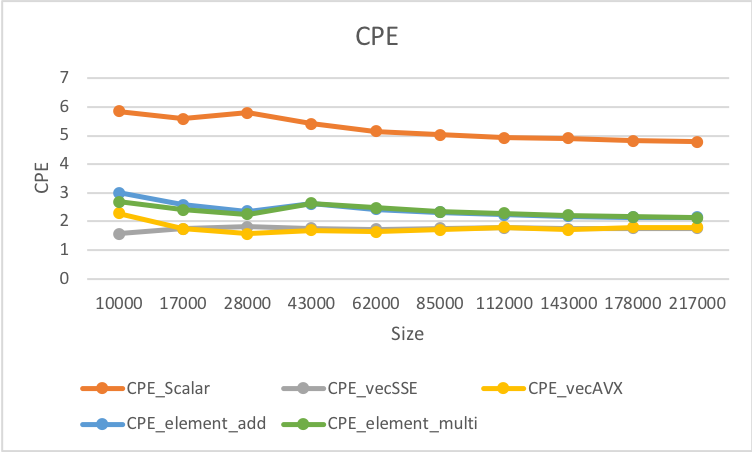
Part 2.

2a.



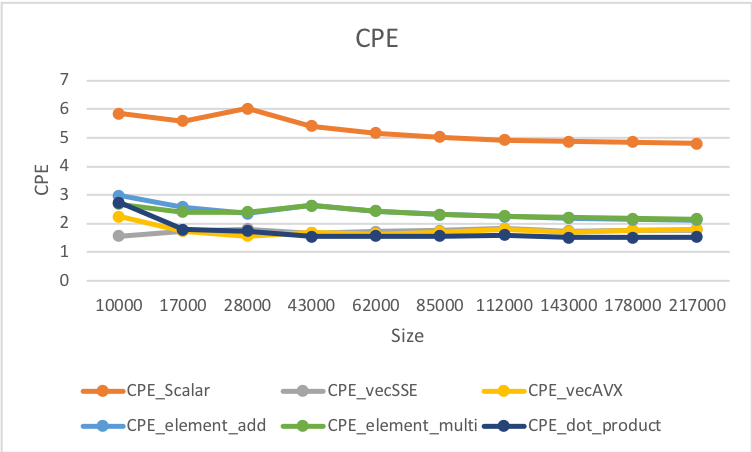
This kind of match what I expected since instead of using solely vectorized code, they use intrinsic to calculate the distances.

2b.



Yes, the above is the CPE.

2c.

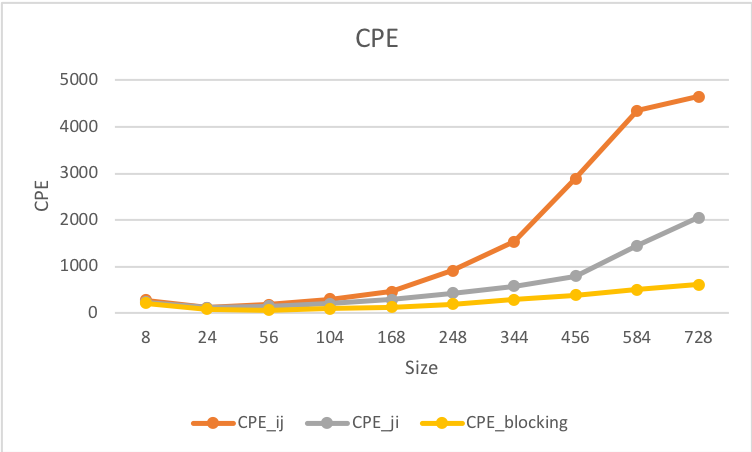


2d.

Intrinsic method is more beneficial in performance compared to vectorization, such as dot product and distance. As for programmability, intrinsic is easier than doing vectorization. When we want to increase the unrolling factor, the more code we will need to write. As we can see that the CPE of the dot product of using intrinsic is the lowest in the graph, which is also lower than vectorized dot product. As for distance calculation, it acts the same way. Using intrinsic (128 bits), the CPE is also lower. The larger the unrolling factor, the difference will be obvious; however, the performance of intrinsic is more efficient overall.

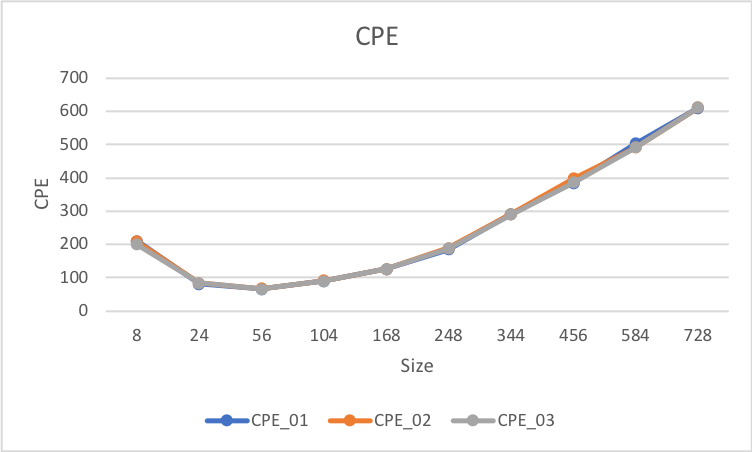
Part 3.

3a.



From the graph, we found that using intrinsic blocking method has the lowest CPE compared to others.

3b.



In the graph, doing 01, 02, or 03 does not make a big difference.