A.

Test 1

Test	Sequential	Parent Wait	Busy-wait	MultiThread w/Sem
Time (ms)	182ms	183ms	0ms	0ms
Product	9,449	7,007	2,458	2,458

Test 2

	Sequential	Parent Wait	Busy-wait	MultiThread w/Sem
Time (ms)	273ms	93ms	9ms	10ms
Product	9,672	7,639	1,092	1,092

Test 3

	Sequential	Parent Wait	Busy-wait	MultiThread w/Sem
Time (ms)	321ms	90ms	7ms	12ms
Product	9,296	318	2,385	2,385

Test 4

	Sequential	Parent Wait	Busy-wait	MultiThread w/Sem
Time (ms)	366ms	185ms	184ms	185ms
Product	4,149	1,063	8,507	1063

Test 5

	Sequential	Parent Wait	Busy-wait	MultiThread w/Sem
Time (ms)	369ms	94ms	92ms	130ms
Product	4,149	277	7,297	277

Test 6

	Sequential	Parent Wait	Busy-wait	MultiThread w/Sem
Time (ms)	365ms	96ms	88ms	112ms
Product	4,149	3,502	7,626	3070

B.

I found this homework assignment to be interesting on how multi-threading can help to increase the time it takes for a process to complete. We can see that in almost every case regular sequential processing took the most time. Since the machine that was used is only 4 Cores with 1 thread per core, we can see that the speedup when we run 8 threads is not optimal and does not fully benefit from running more threads. Test 4 makes sense as it is approximately gaining a 2 times speed up where there are 2 threads and has to run to the end of the array.