Part I: Implementation:					
Requirement	C	Algorithm	Linux	gcc compiler	pthread.h
	Yes	Yes	Yes	Yes	yes

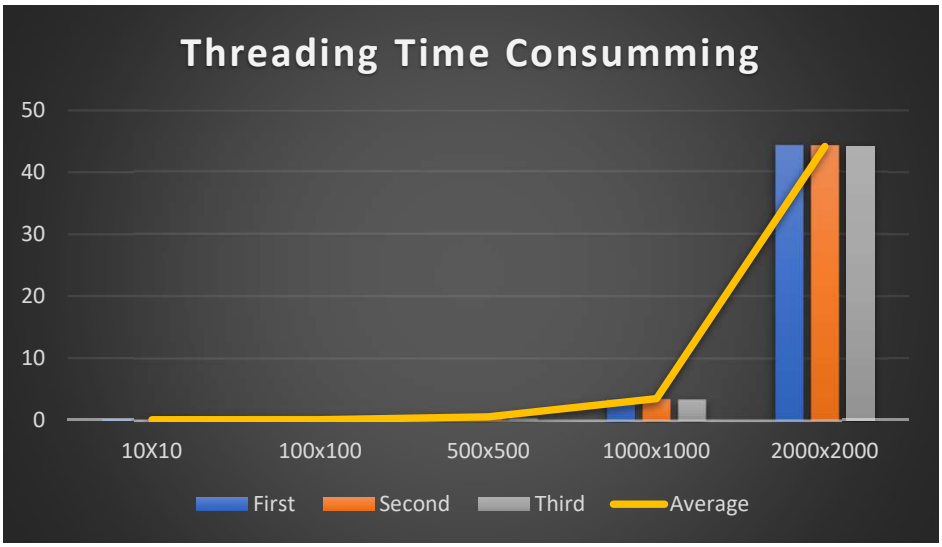
Part II: Experimentation:					
Requirement	dynamic sizes	malloc()	free()	random()	int argc, char *argv[]
	Yes	Yes	yes	yes	yes

Part III: Multi Threading run

Instruction: Use five source code with define each max index (MAX), and number of thread (n) 10, 100, 500, 1000, 2000. Name the program repestively: thread10, thread100, thread500, thread1000, thread2000. Execute each program 3 times.

Command use:	Command	Repeat Times
	sudo nano thread10.c	5
	gcc -pthread -o thread10 thread10.c	5
	time ./thread10	5
Total execution time (real time) are being record. **Measure: second**		

Part III: Multi Threading table					
Program	thread10	thread100	thread500	thread1000	thread2000
Dimension	10X10	100x100	500x500	1000x1000	2000x2000
First	0.002	0.028	0.515	3.385	44.243
Second	0.001	0.029	0.52	3.493	44.193
Third	0.002	0.026	0.419	3.406	44.136
Average	0.001666667	0.027666667	0.484666667	3.428	44.19066667

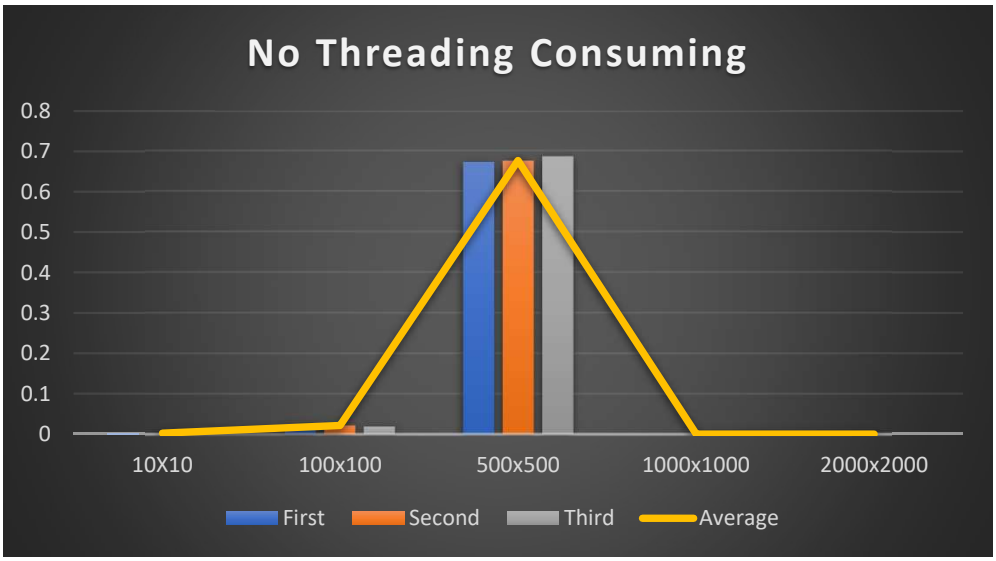


Part III: No Threading (Assignment1)

Instruction: Use five source code with define each max index(MAX) 10, 100, 500, 1000, 2000. Name the program repestively: thread10, thread100, thread500, thread1000, thread2000. Execute each program 3 times. Change

Command use	Command	Execute Times
	sudo nano matrix10.c	5
	gcc -o matrix10 matrix10.c	5
	time ./matrix10	5
Total execution time (real time) are being record. **Measure: second**		

Part III: No Threading table(Assignment1)					
Program	matrix10	matrix100	matrix500	matrix1000	matrix2000
Dimension	10X10	100x100	500x500	1000x1000	2000x2000
First	0.002	0.021	0.672	Sefmentation fault	Sefmentation fault
Second	0.001	0.022	0.675	Sefmentation fault	Sefmentation fault
Third	0.001	0.019	0.686	Sefmentation fault	Sefmentation fault
Average	0.001333333	0.020666667	0.677666667	Stack overflow	Stack overflow



Compare

Avg.	Multi-Threading (M.T)	No Thread(N.T.)	Result
10 x 10	0.001666667	0.001333333	N.T is better
100 x 100	0.027666667	0.020666667	N.T is better
500 x 500	0.484666667	0.677666667	M.T is better
1000 x 1000	3.428	Stack overflow	M.T is better
2000 x 2000	44.19066667	Stack overflow	M.T is better

Conclusion: in small calulation, no threading run a little bit better. But it is definitely not effcient for mdium and large calculation. Overall threading is much better.