**Evaluation Report**

Purpose of the experiment : To find a way to reduce computational time for complex problems, using multi-threading.

Experimental Setup : In the assignment, I have created multiple threads to form the image. Initially, I get the height and number and number of threads from the user and store in separate parameters. On dividing the height of the image by the number of threads, I get the threshold for each thread. So first thread will run for the fixed threshold, then the second starts from where the first stopped, and so on. On dividing, I did not always get a whole number so I used the ceil() function to help fix this issue and set the end limit of the final thread equal to the height of the image. So all the threads ran continuously one after the other to form the image.

Describe why you chose the threading library and implementation : It is implemented using C++ threads because I had already worked with this before and found it easier compared to other libraries.

Graph of Execution Time :

Shape of the curves : Curve A and B are declining curves. Although curve B increases a little at thread 5. The curves have different shapes because curve A and B have different amount of work put in for each pixel and have different start and end points.

Anomalous Results : For curve A there were no anomalous results as the execution time get reducing on increasing the number of threads. But, for curve B, at 5 threads the execution time increased from 4.096s (time for 4 threads) to 4.419s and then went back down to 3.818s for 10 threads.