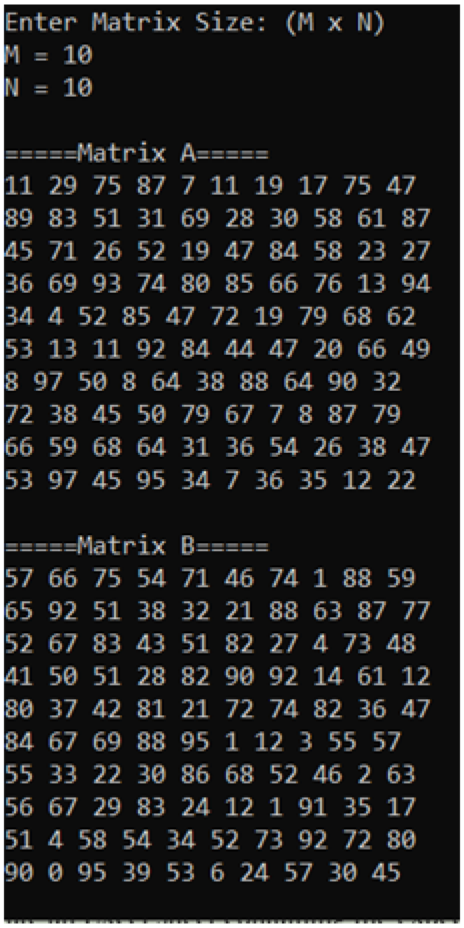
IJOSHUA MANUEL LOUISE KEMPIS 4 BS CpE March 13, 2023

ENGG 126 Project 2 Third Progress Report

Matrix Multiplication Speed Test

Implementation of Threads and Comparing Speed results

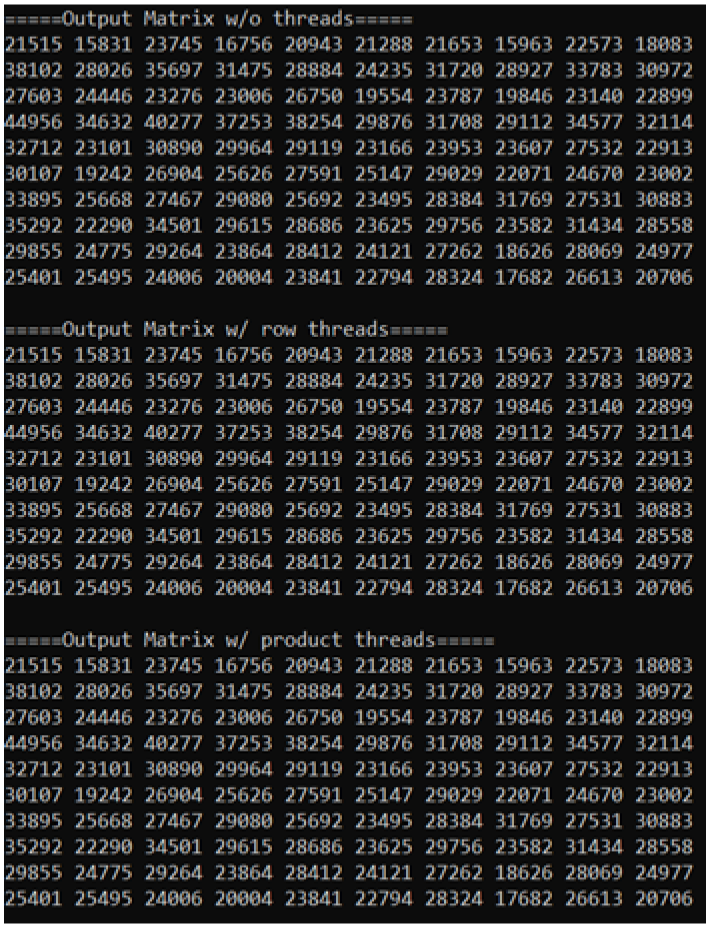
The function to implement threads in the multiplication process of the matrices has been implemented using the threads_row() and threads_prod() functions. This goes with the implementation of the multiplication operation functions matrixMult_row() and matrixMult_prod() which came from the original matrixMult() function from the previous implementation of the code. The results of the current code contains these output screenshots:



```
Enter Matrix Size: (M x N)
M = 10
N = 10

=====Matrix A=====
11 29 75 87 7 11 19 17 75 47
89 83 51 31 69 28 30 58 61 87
45 71 26 52 19 47 84 58 23 27
36 69 93 74 80 85 66 76 13 94
34 4 52 85 47 72 19 79 68 62
53 13 11 92 84 44 47 20 66 49
8 97 50 8 64 38 88 64 90 32
72 38 45 50 79 67 7 8 87 79
66 59 68 64 31 36 54 26 38 47
53 97 45 95 34 7 36 35 12 22

=====Matrix B=====
57 66 75 54 71 46 74 1 88 59
65 92 51 38 32 21 88 63 87 77
52 67 83 43 51 82 27 4 73 48
41 50 51 28 82 90 92 14 61 12
80 37 42 81 21 72 74 82 36 47
84 67 69 88 95 1 12 3 55 57
55 33 22 30 86 68 52 46 2 63
56 67 29 83 24 12 1 91 35 17
51 4 58 54 34 52 73 92 72 80
90 0 95 39 53 6 24 57 30 45
```



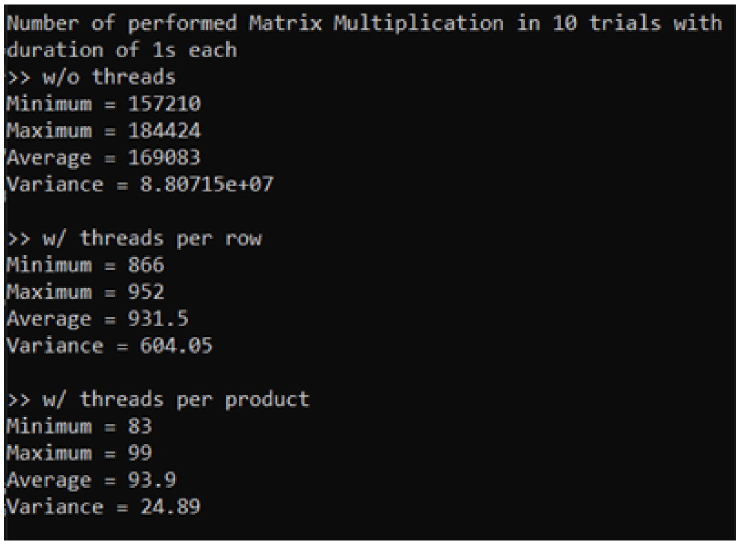
```
=====Output Matrix w/o threads=====
21515 15831 23745 16756 20943 21288 21653 15963 22573 18083
38102 28026 35697 31475 28884 24235 31720 28927 33783 30972
27603 24446 23276 23006 26750 19554 23787 19846 23140 22899
44956 34632 40277 37253 38254 29876 31708 29112 34577 32114
32712 23101 30890 29964 29119 23166 23953 23607 27532 22913
30107 19242 26904 25626 27591 25147 29029 22071 24670 23002
33895 25668 27467 29080 25692 23495 28384 31769 27531 30883
35292 22290 34501 29615 28686 23625 29756 23582 31434 28558
29855 24775 29264 23864 28412 24121 27262 18626 28069 24977
25401 25495 24006 20004 23841 22794 28324 17682 26613 20706

=====Output Matrix w/ row threads=====
21515 15831 23745 16756 20943 21288 21653 15963 22573 18083
38102 28026 35697 31475 28884 24235 31720 28927 33783 30972
27603 24446 23276 23006 26750 19554 23787 19846 23140 22899
44956 34632 40277 37253 38254 29876 31708 29112 34577 32114
32712 23101 30890 29964 29119 23166 23953 23607 27532 22913
30107 19242 26904 25626 27591 25147 29029 22071 24670 23002
33895 25668 27467 29080 25692 23495 28384 31769 27531 30883
35292 22290 34501 29615 28686 23625 29756 23582 31434 28558
29855 24775 29264 23864 28412 24121 27262 18626 28069 24977
25401 25495 24006 20004 23841 22794 28324 17682 26613 20706

=====Output Matrix w/ product threads=====
21515 15831 23745 16756 20943 21288 21653 15963 22573 18083
38102 28026 35697 31475 28884 24235 31720 28927 33783 30972
27603 24446 23276 23006 26750 19554 23787 19846 23140 22899
44956 34632 40277 37253 38254 29876 31708 29112 34577 32114
32712 23101 30890 29964 29119 23166 23953 23607 27532 22913
30107 19242 26904 25626 27591 25147 29029 22071 24670 23002
33895 25668 27467 29080 25692 23495 28384 31769 27531 30883
35292 22290 34501 29615 28686 23625 29756 23582 31434 28558
29855 24775 29264 23864 28412 24121 27262 18626 28069 24977
25401 25495 24006 20004 23841 22794 28324 17682 26613 20706
```

Given 10 trials with a duration of 1s each, the number of Matrix Multiplication operations were tested and the output for the Minimum, Maximum, Mean, and Variance results were provided.

These data is in relation to the speed at which the program was able to perform the multiplication operation following the implementation of the new matrixMult functions as well as the speedTest() function



```
Number of performed Matrix Multiplication in 10 trials with
duration of 1s each
>> w/o threads
Minimum = 157210
Maximum = 184424
Average = 169083
Variance = 8.80715e+07

>> w/ threads per row
Minimum = 866
Maximum = 952
Average = 931.5
Variance = 604.05

>> w/ threads per product
Minimum = 83
Maximum = 99
Average = 93.9
Variance = 24.89
```

Conclusion

The project currently has its updated code implementation with a working Matrix generator and is capable of conducting the matrix multiplication operation as well as the implementation of threads and comparison between results of the speedTest() function. These outputs record various instances of Multiplication trials (w/o threads, w/ threads per row, & w/ threads per product) given a certain duration time. The program and project are around 90% complete and are currently in its second revision stage.

Task	**Description**	**Status**
Project Research	Search suitable components for the survey project	DONE
Project Outline	Create a draft document as a guide for the project paper	DONE
Initial Implementation	A C++ code implementation was created which acts as the foundation of the project	PARTIAL
Revision I & II	Following each progress report, provide the necessary corrections and apply feedback & revisions	PARTIAL
Finalize Paper	Finish the paper and draw insights and conclusions on the Project	ONGOING