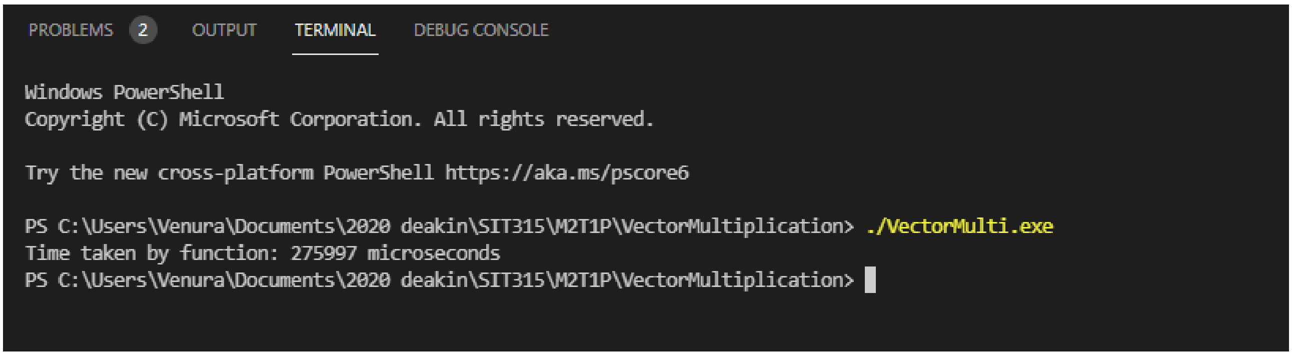
TaskM2.T1P: Parallel Matrix Multiplication Documentation

The sequential program I made for the matrix multiplication was able to achieve an execution of 275997 microseconds. The matrix size was 100000000.

.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Venura\Documents\2020 deakin\SIT315\M2T1P\VectorMultiplication> ./VectorMulti.exe
Time taken by function: 275997 microseconds
PS C:\Users\Venura\Documents\2020 deakin\SIT315\M2T1P\VectorMultiplication>
```

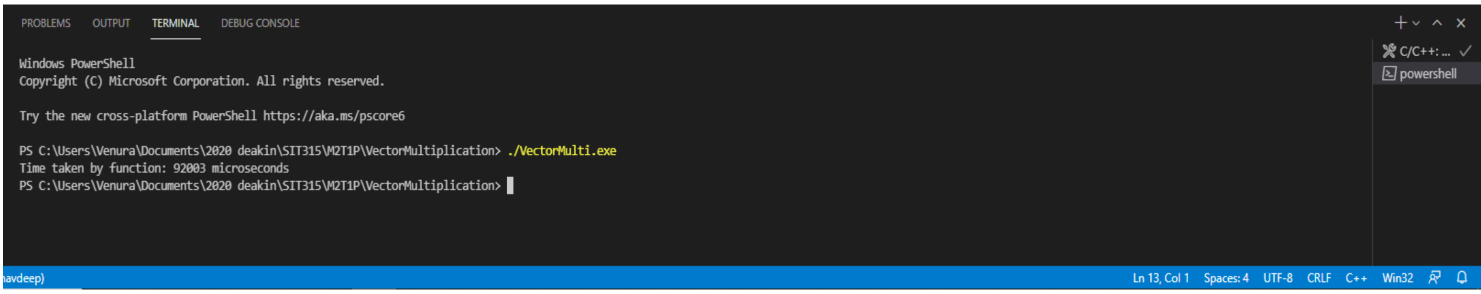
Decomposition of parallelism of the Matrix Multiplication:

I will attempt to use data output decomposition in the parallelism of matrix multiplication program. The partitioning will be done on the rows of the matrix.

Decomposition tasks

1. The partitioning will begin with creating a subtask that can partition the rows which can be computed by multiple threads independently.
2. Create Variable called proportion which will be used to manipulate the start point of an iteration in order to segment the rows.
3. The end point for each segmented row will be calculated by adding the proportion variable on top of the start point of the segmented row.

Performance Evaluation



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Venura\Documents\2020 deakin\SIT315\M2T1P\VectorMultiplication> ./VectorMulti.exe
Time taken by function: 92003 microseconds
PS C:\Users\Venura\Documents\2020 deakin\SIT315\M2T1P\VectorMultiplication>
```

A equivalent matrix size of 100000000 for the parallel program took 92003 seconds this was atleast **3 times faster** than the sequential program.

However for a smaller matrix size such as the 1000 size matrix the sequential program proved to be faster. The performance was heavily dependent on the data-set size.