Description: Create a program that will utilize different variations of matrix multiplication in order to show the effects of cache and memory utilization

This assignment gives students the opportunity to see the difference in speed between normal matrix multiplication, transposed matrix multiplication, block multiplication, and multiple children processing each block of the block multiplication created by the fork command.

The size of each matrix, to obtain valid output, is always a power of 2. For example, size 4 = 2^2, 9 = 3^2, 16 = 4^2. Any value that does not meet these specifications could cause unintended errors.

The block size of the matrices to be used in the block multiplication was based on numbers that were 1) the square root of the matrix size 2) matrix size/8 3) Matrix size/4 4) Matrix size/8. Therefore all inputs used to collect data met each of those requirements.

The table below represents the matrix multiplication which does not utilize any forks/children. Values beyond 2304 took exceptionally long so those greater values are not contained in this report.



The table below represents the matrix multiplication which utilizes forks/children in order to perform the multiplication of each block at the same time. Semaphores have been implemented in the program in order to prevent multiple processes from writing to the same element of Matrix C at the same time. Values beyond 7744 took exceptionally long so those greater values are not contained in this report.

The values seem to show that the closer the block size is to n, the faster the computations. However, I think with extremely large values, beyond the scope and capabilities of this project, there may be a case where the speed of computation is inversely proportional to the size of the matrix

