Assignment - A2 41403

- Title: Parallel computing using CUDA

- Problem statement: vector and matrix operations
  design parallel algorithm to
  - add two large vectors
  - multiply vector & matrix
  - multiply two NxN arrays using N^2 process

- Objectives: learn parallel computing using CUDA & parallel decomposition of a problem

- Outcome: decomposed problem into sub-problems learned how to use GPUs solved problems using threads on GPU cores.

- S/w & H/w: 64 bit CPU, 4GB RAM, CUDA toolkit, Nvidia, NVCC compiler Google colab.

- Theory: Dividing a computation into smaller computations & assigning tem to different processors for parallel execution are two key steps in the design of parallel algorithm
  The process of dividing a computation in smaller parts some or all which may potentially be executed in parallel is called decomposition.

Tasks are programmer defined units of computation into which the main computation is subdivided by means of decomposition.
Simultaneous execution of multiple tasks is the key to reducing time required to solve the entire problem.
Tasks can be of arbitrary size, but once defined they are regarded as indivisible units of computation

The tasks into which a problem is decomposed may not all be the same size

In addition of two vectors, we have to add ith element from 1st array with ith element of 2nd array to get ith element of result

This allocation can be allocated to a distinct thread. The same thing is done for vector product matrix
Using CUDA, vectors can be added using
- n blocks, 1 thread/block
- 1 block n threads
- m blocks n threads/block

- Similarly the product of vector & matrix will result in a 1xn vector containing the

result of multiplication
- The product of two matrices will resut in matrix of resultant

CUDA Kernels & Threads
The fundamental part of a CUDA code is the Kernel program.
kernel is the function that can be executed in parallel in the GPU device.
A CUDA Kernel is executed by an array of CUDA threads. All threads run the same code.
Each thread has an id that it uses to compute memory address & make control decisions. CUDA organizes thousands of threads into a hierarchy of a grid of thread blocks. A grid is a set of thread blocks that can be processed on a device in parallel.
A Thread block is a set of concurrent threads that can cooperate among themselves through synchronization barriers & access to a stored memory space private to the block. Each thread is given an ID unique within the block each block has a unique ID within a grid.

- ~~code~~

- conclusion: successfully implemented & executed vector & matrix operations parallely using CUDA.