# HPCA Programming Assignment Part B
# Checkered Matrix Multiplication in CUDA

**Overview:**
In this assignment, you need to write CUDA version of the CMM program that you had optimized and multi-threaded for CPU in part A.

**Deadline: Friday, 23$^{rd}$ November, 2021**

**Deliverables:**
Similar to part A, you are required to submit a report along with your code. The report should include details about the approach you considered while implementing and where you ran you program (GPGPU-Sim/GPU). Optionally, you can also include what further optimizations are possible to reduce the runtime of your implementation.

**What/how to submit:**
Clone the GitLab repository on your machine and follow its README.md for instructions on how to compile and run the programs. All your code has to be implemented in "header/". DO NOT MODIFY main.cu.

Gitlab Repo: https://gitlab.com/shweta_pandey/hpca-assignment-2021/-/tree/main

Submit your assignment as a tar.gz file named as per the last five digits of your SR number (e.g., 15964.tar.gz). It should contain two directories – i.e., "headers/" that would contain your implementation and "reports/". Name your reports as "reports/Report_PartB.pdf".

**Where to run/test your CUDA program:**
You all would have gotten access to the GPU server of the department. You may access GPUs in that server to run your program. However, expect the GPUs to be very busy during the end of the semester. Therefore, we recommend using GPGPU-sim (see below for installation instructions). This is a software that runs on the CPU but emulates a GPU. It runs CUDA application out-of-the-box. You can install this simulator in your local machine and do your homework.

**Prerequisites:**
In order to compile CUDA code (whether for a GPU or GPGPU-Sim), you will need to install the CUDA compiler (nvcc). You can download and install this from the official NVIDIA website (https://developer.nvidia.com/cuda-11.0-download-archive). You do NOT need to install the driver if you plan to use GPGPU-sim. GPGPU-Sim currently works with CUDA versions 4.5 - 11.0.

**Instructions for using GPGPU-sim:**

- Clone GPGPU-sim using the following link. https://github.com/gpgpu-sim/gpgpu-sim_distribution
- Run your code for the SM7_TITANV configuration. You can find the configuration file at : gpgpu-sim_distribution/configs/tested-cfgs/
- You can use GPGPU-sim profiler to optimize the performance your CUDA application. If you are running on the GPU hardware, look for NSight and/or NVprof tools.