**Class:** Final Year (Computer Science and Engineering)

**Year:** 2021-22 **Semester:** 1 **Date:** 21/11/2021

**Course:** High Performance Computing Lab

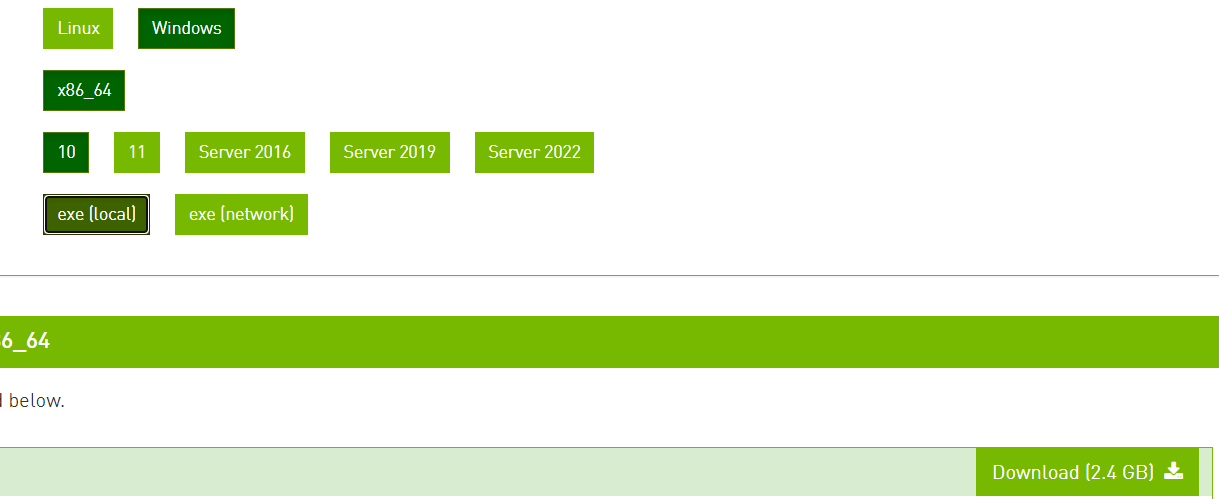
**Practical No. 7**

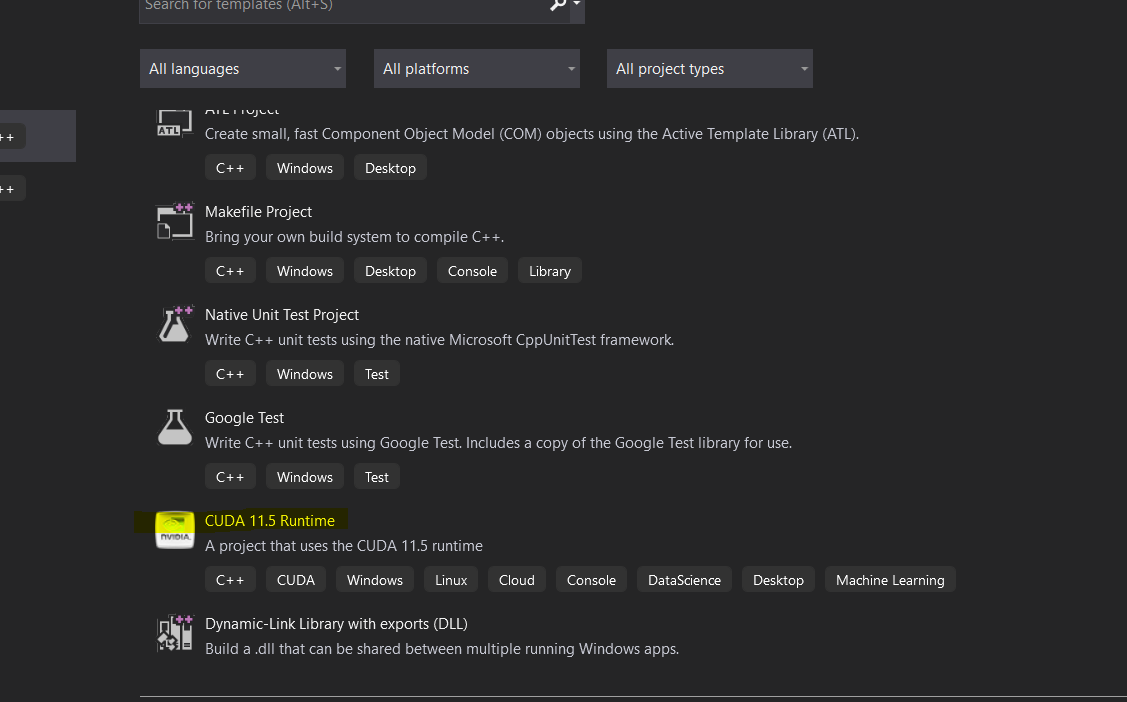
**Exam Seat No:**

2018BTECS00071 – Saurabh Makarand Narkhede

**Problem Statement 1:**

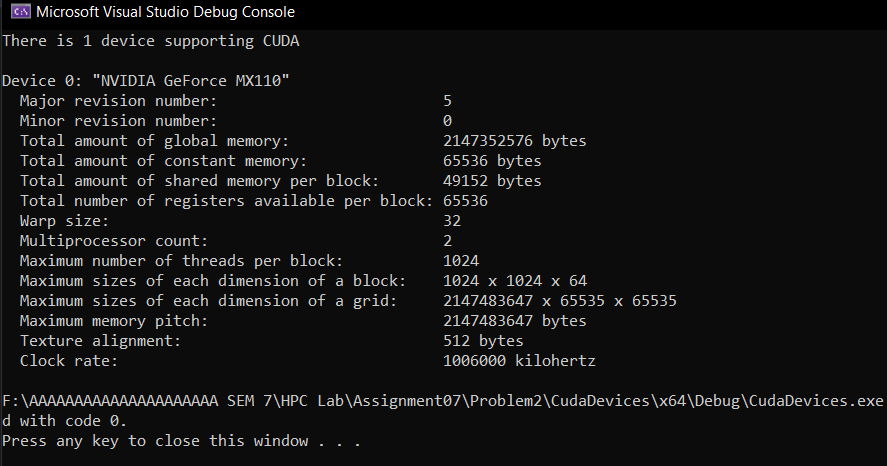
Setup environment requirements for CUDA C programs.



****

**Problem Statement 2:**

Execute the attached Program 1, and understand the output.

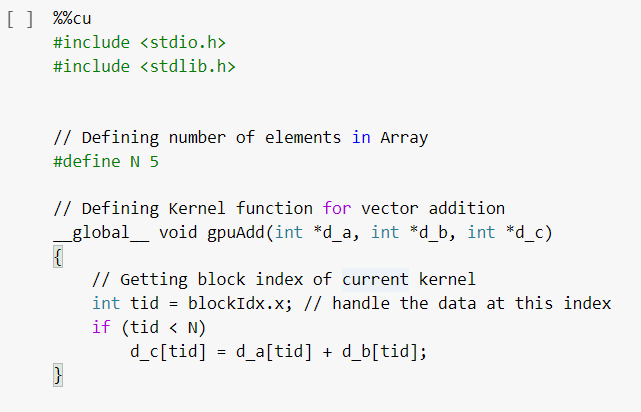


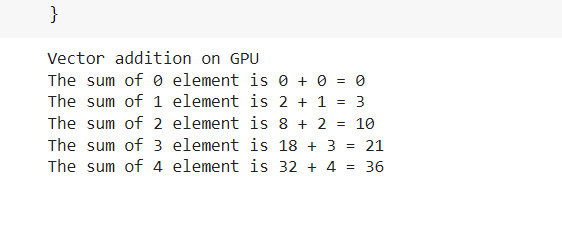
The information of NVDIA GeForce MX110 GPU given. Where total amount of global memory, constant memory, shared memory per Block is Displayed. Also other information about grid & block and threads per block is given.

**Gives us a small idea of how the thousands of threads will parallelly execute.**

**Problem Statement 3:**

Write a CUDA C program to perform the addition of two vectors of arbitrary size(Dynamic Array).





Here’s the structure of the GPU main function:

* It starts with defining arrays and pointers for host and device. The device pointers are allocated memory using the cudaMalloc function.
* The arrays, which are to be passed to the kernel, are copied from the host memory to the device memory by using the cudaMemcpy function.
* The kernel is launched by passing the device pointers as parameters to it. If you see the values inside the kernel launch operator, they are **N** and **1**, which indicate we are launching **N** blocks with one thread per each block.
* The answer stored by the kernel on the device memory is copied back to the host memory by again using the cudaMemcpy, but this time with the direction of data transfer from the device to the host.
* And, finally, memory allocated to three device pointers is freed up by using the cudaFree function.