Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

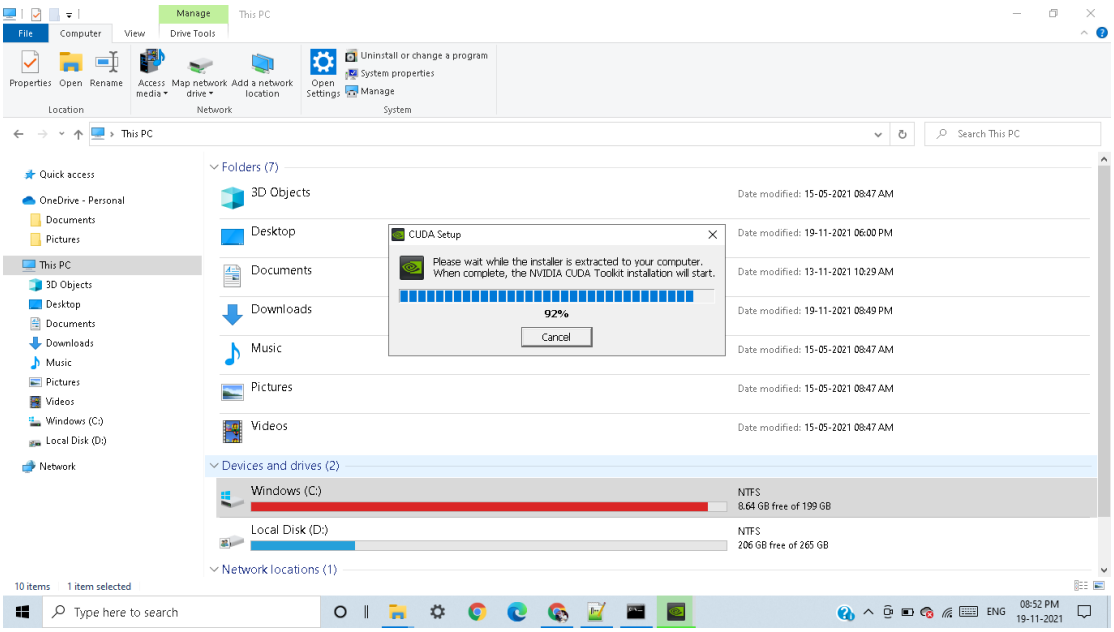
Practical No. 7

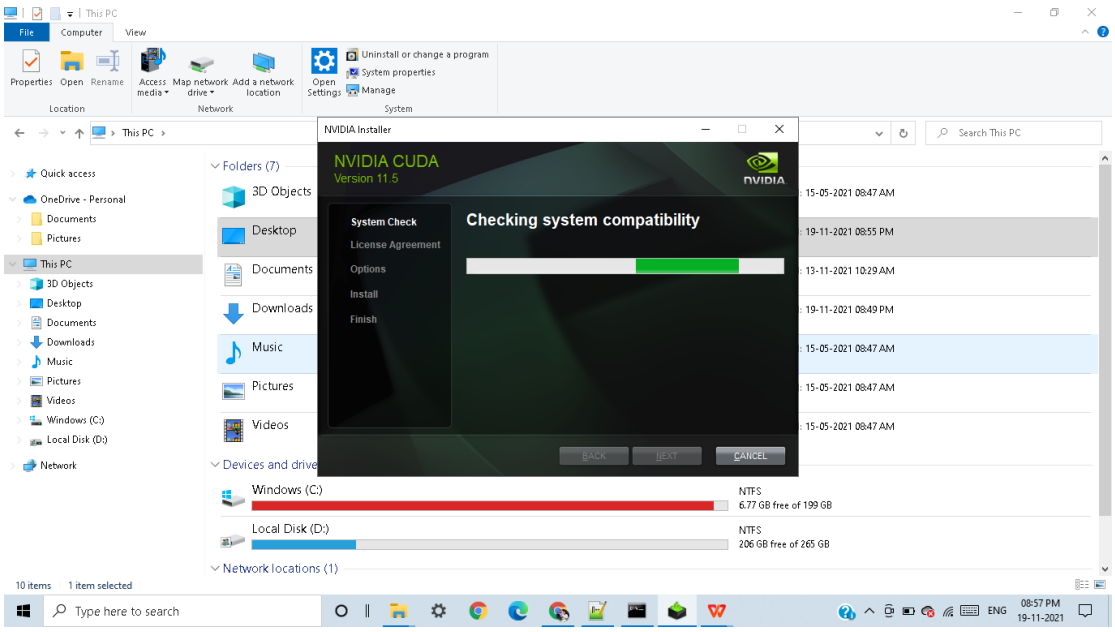
Study and Implementation of CUDA C basic functions.

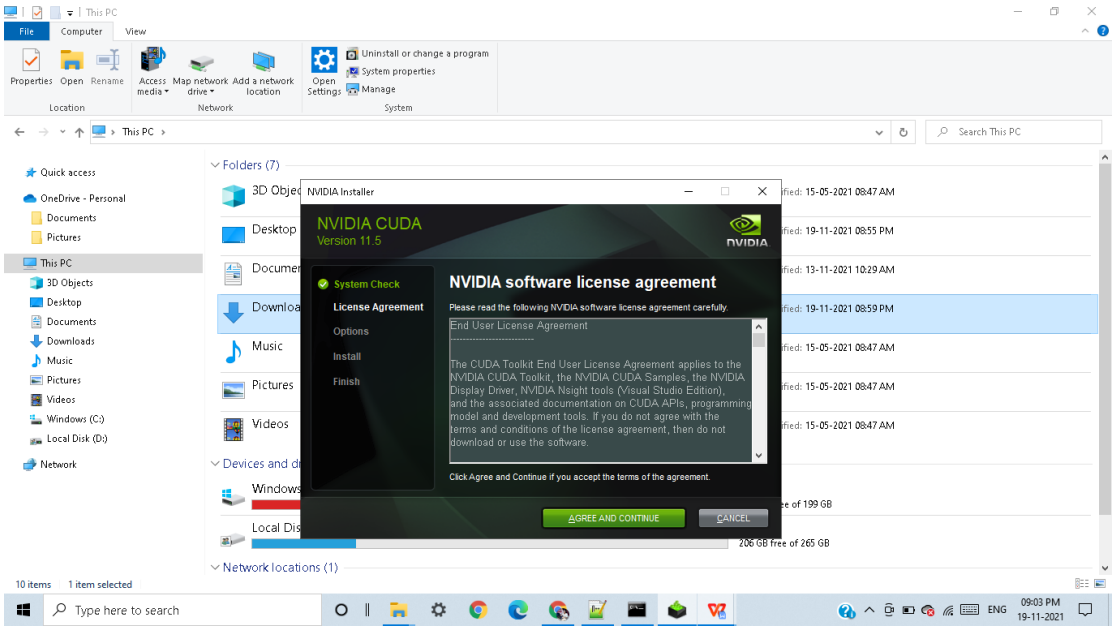
NAME: SHWETA NANDKUMAR ARBUNE
PRN: 2019BTECS00205

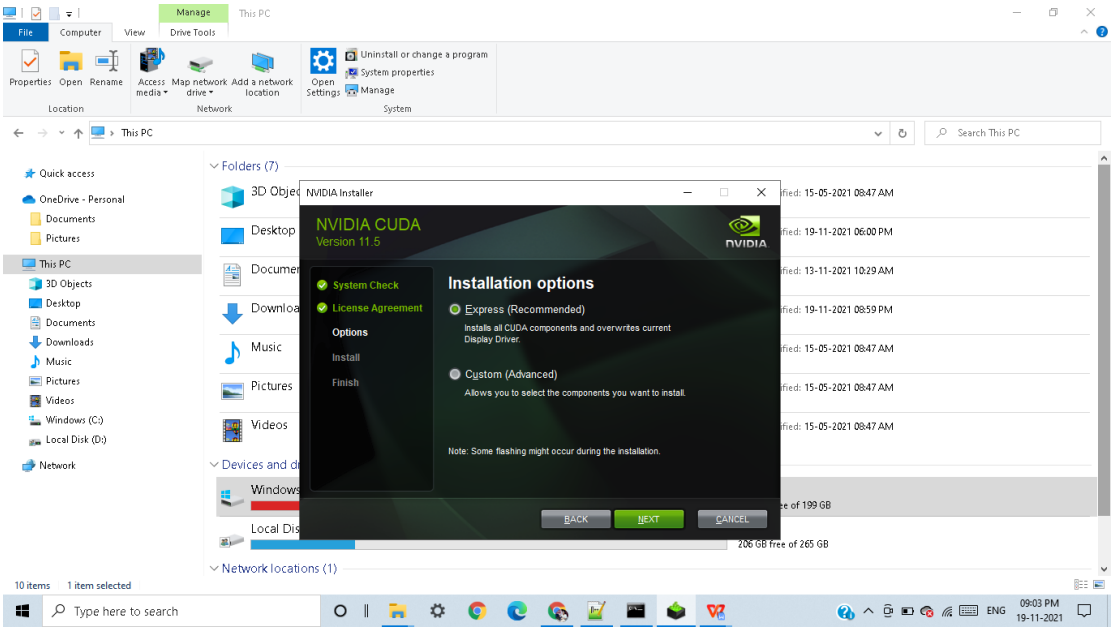
1) Setup the environment requirements, for execution of CUDA C programs.

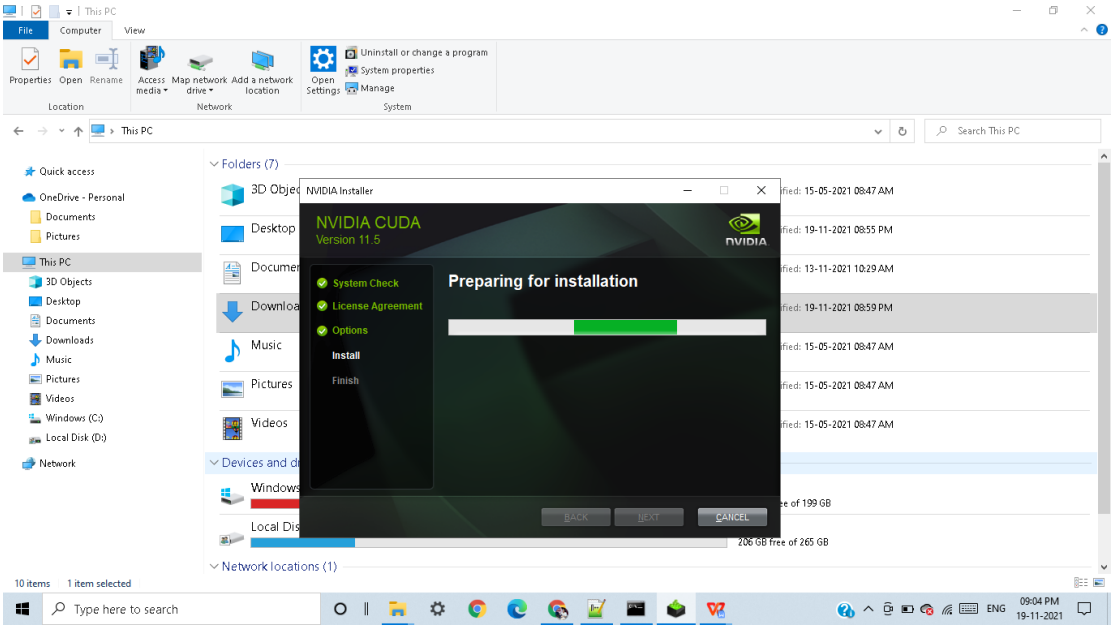
A. CUDA Toolkit Installation :



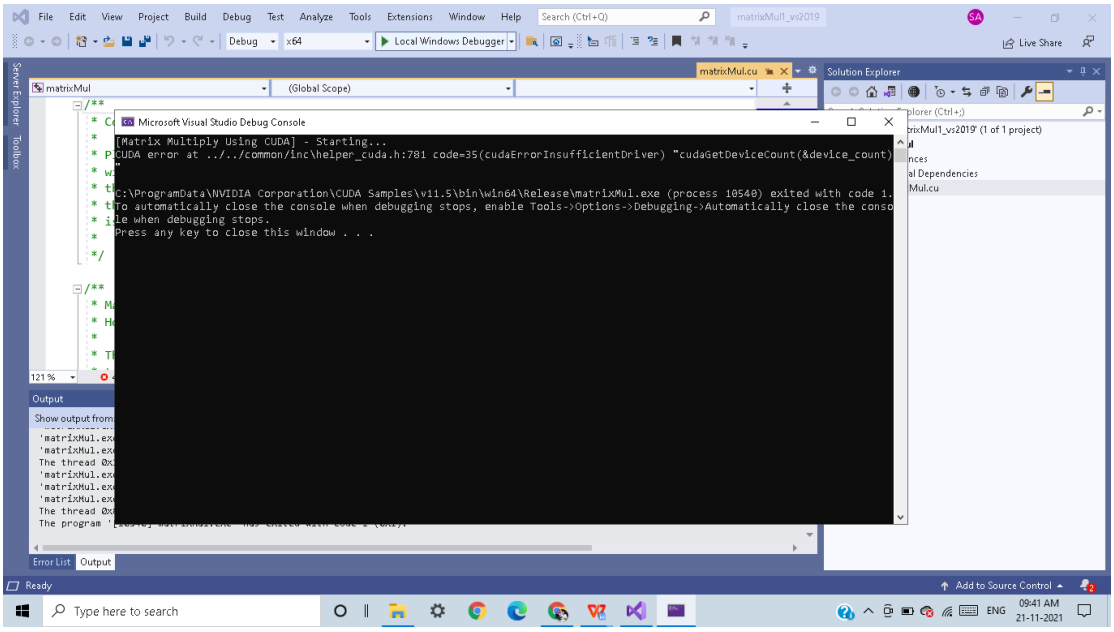

NVIDIA Installer
NVIDIA CUDA
Version 11.5
NVIDIA
System Check
License Agreement
Options
Install
Finish
Checking system compatibility
BACK
NEXT
CANCEL
This PC
Folders (7)
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
NTFS
6.77 GB free of 199 GB
Local Disk (D:)
NTFS
206 GB free of 265 GB
Network locations (1)
10 items
1 item selected
Type here to search
ENG
08:57 PM
19-11-2021


NVIDIA Installer
NVIDIA CUDA
Version 11.5
NVIDIA
System Check
License Agreement
Options
Install
Finish
NVIDIA software license agreement
Please read the following NVIDIA software license agreement carefully.
End User License Agreement
The CUDA Toolkit End User License Agreement applies to the NVIDIA CUDA Toolkit, the NVIDIA CUDA Samples, the NVIDIA Display Driver, NVIDIA Nsight tools (Visual Studio Edition), and the associated documentation on CUDA APIs, programming model and development tools. If you do not agree with the terms and conditions of the license agreement, then do not download or use the software.
Click Agree and Continue if you accept the terms of the agreement.
AGREE AND CONTINUE
CANCEL
This PC
Folders (7)
Network locations (1)
10 items
1 item selected
Type here to search
ENG
09:03 PM
19-11-2021

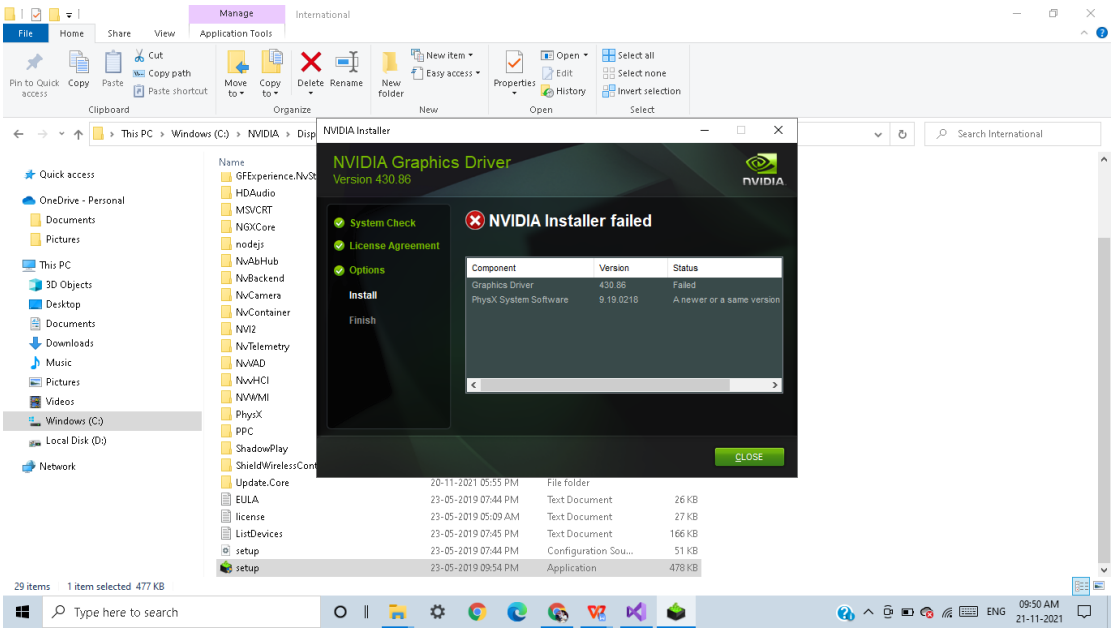




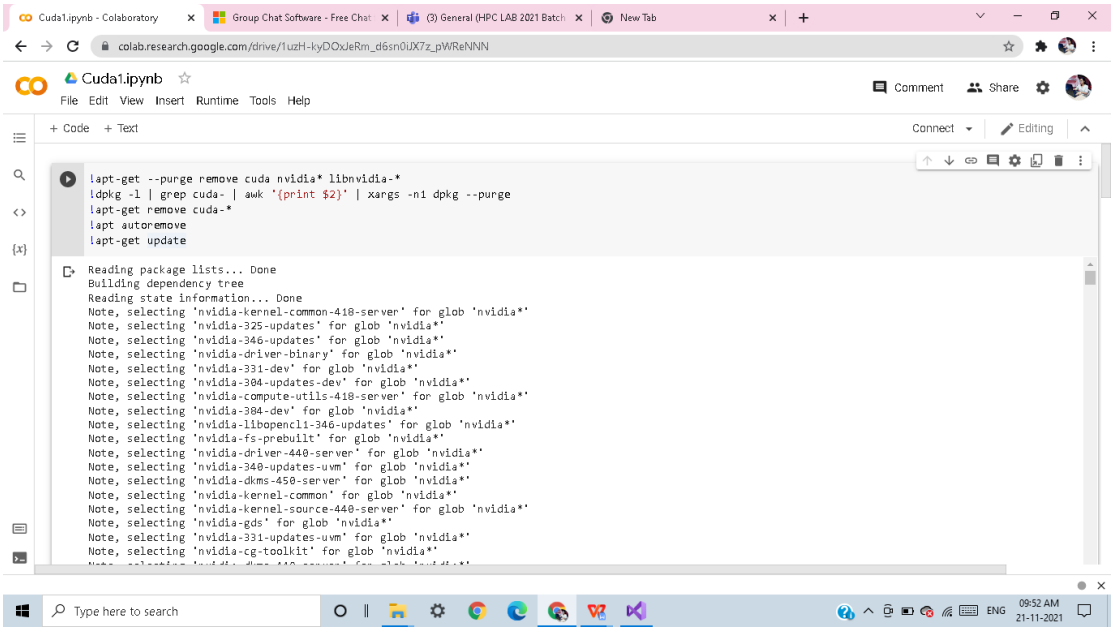
B. Showing Error of CudaDrivers:

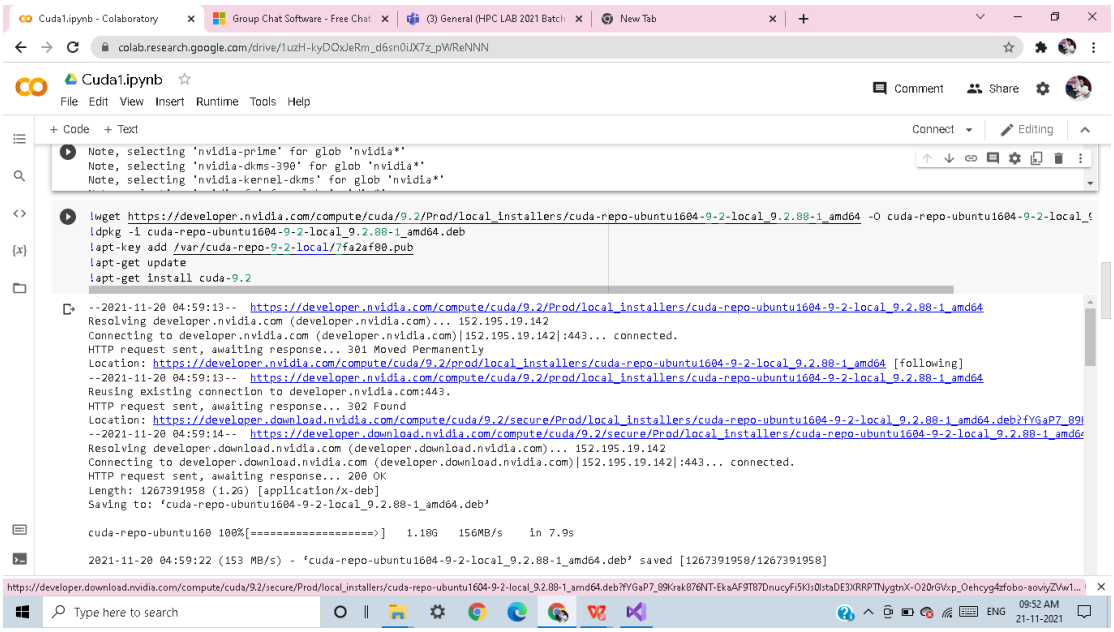


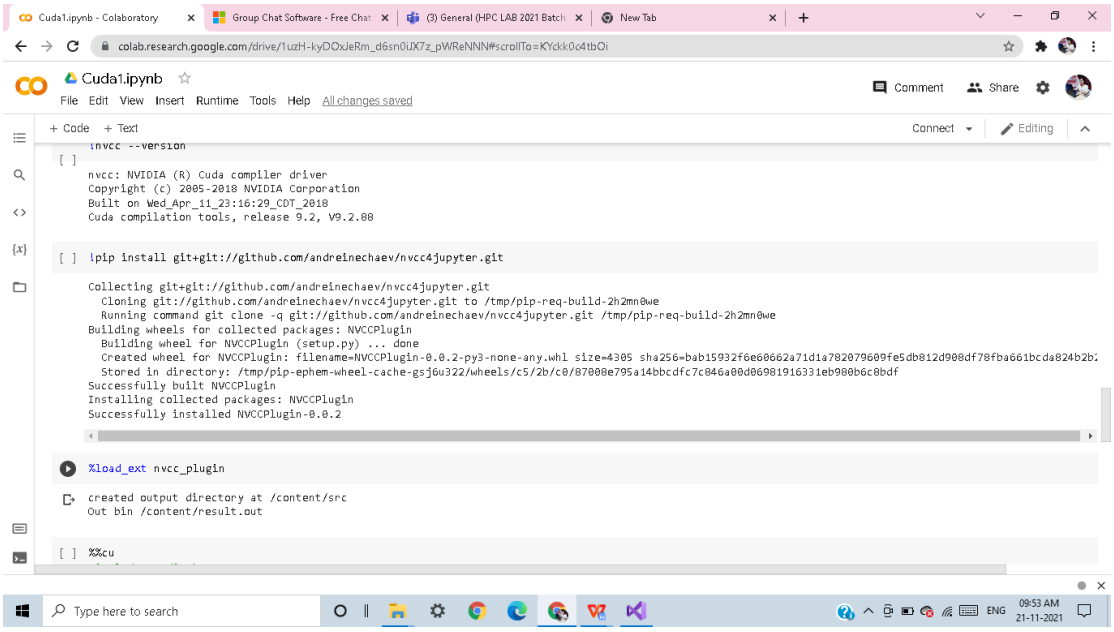
C. CUDA Drivers Installation Failed :



D. CUDA Setup on Google colab:


Cuda1.ipynb - Colaboratory
Group Chat Software - Free Chat
(3) General (HPC LAB 2021 Batch
New Tab
colab.research.google.com/drive/1uzH-kyDOxJeRm_d6sn0iJX7z_pWReNNN
Cuda1.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
Connect
Editing
!apt-get --purge remove cuda nvidia* libnvidia-*
!dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 dpkg --purge
!apt-get remove cuda-*
!apt autoremove
!apt-get update
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'nvidia-kernel-common-418-server' for glob 'nvidia*'
Note, selecting 'nvidia-325-updates' for glob 'nvidia*'
Note, selecting 'nvidia-346-updates' for glob 'nvidia*'
Note, selecting 'nvidia-driver-binary' for glob 'nvidia*'
Note, selecting 'nvidia-331-dev' for glob 'nvidia*'
Note, selecting 'nvidia-304-updates-dev' for glob 'nvidia*'
Note, selecting 'nvidia-compute-utils-418-server' for glob 'nvidia*'
Note, selecting 'nvidia-304-dev' for glob 'nvidia*'
Note, selecting 'nvidia-libopencl1-346-updates' for glob 'nvidia*'
Note, selecting 'nvidia-fs-prebuilt' for glob 'nvidia*'
Note, selecting 'nvidia-driver-440-server' for glob 'nvidia*'
Note, selecting 'nvidia-340-updates-uvm' for glob 'nvidia*'
Note, selecting 'nvidia-dkms-450-server' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-common' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-source-440-server' for glob 'nvidia*'
Note, selecting 'nvidia-gds' for glob 'nvidia*'
Note, selecting 'nvidia-331-updates-uvm' for glob 'nvidia*'
Note, selecting 'nvidia-cg-toolkit' for glob 'nvidia*'
Type here to search
ENG
09:52 AM
21-11-2021


Cuda1.ipynb - Colaboratory
Group Chat Software - Free Chat
(3) General (HPC LAB 2021 Batch
New Tab
colab.research.google.com/drive/1uzH-kyDOxJeRm_d6sn0iJX7z_pWReNNN
Cuda1.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
Connect
Editing
Note, selecting 'nvidia-prime' for glob 'nvidia*'
Note, selecting 'nvidia-dkms-390' for glob 'nvidia*'
Note, selecting 'nvidia-kernel-dkms' for glob 'nvidia*'
!wget https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 -O cuda-repo-ubuntu1604-9-2-local_
!dpkg -i cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
!apt-key add /var/cuda-repo-9-2-local/7fa2af80.pub
!apt-get update
!apt-get install cuda-9.2
--2021-11-20 04:59:13-- https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64
Resolving developer.nvidia.com (developer.nvidia.com)... 152.195.19.142
Connecting to developer.nvidia.com (developer.nvidia.com)|152.195.19.142|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://developer.nvidia.com/compute/cuda/9.2/prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 [following]
--2021-11-20 04:59:13-- https://developer.nvidia.com/compute/cuda/9.2/prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64
Reusing existing connection to developer.nvidia.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://developer.download.nvidia.com/compute/cuda/9.2/secure/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb?fYGaP7_89
--2021-11-20 04:59:14-- https://developer.download.nvidia.com/compute/cuda/9.2/secure/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd6
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 152.195.19.142
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|152.195.19.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1267391958 (1.2G) [application/x-deb]
Saving to: 'cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb'
cuda-repo-ubuntu160 100%[==========>] 1.18G 156MB/s in 7.9s
2021-11-20 04:59:22 (153 MB/s) - 'cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb' saved [1267391958/1267391958]
https://developer.download.nvidia.com/compute/cuda/9.2/secure/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb?fYGaP7_89Krak876NT-EkaAF9T87DnucyFi5Kl:0JstaDE3XRRPTNygtnX-O20rGVxp_Oehcyg4zfobo-aovjyZVw1...
Type here to search
ENG
09:52 AM
21-11-2021



2) Execute the attached Program 1, and understand the output.

```
There is 1 device supporting CUDA

Device 0: "Tesla K80"
  Major revision number:                         3
  Minor revision number:                         7
  Total amount of global memory:                 -887947264 bytes
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Multiprocessor count:                          13
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a block:    1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:     2147483647 x 65535 x 65535
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Clock rate:                                    562000 kilohertz

✓ 0s  completed at 11:26 AM
```

Device_Count is basically used to check the GPU device available on system or not, In google colab here 1 device with id 0 is present and the name of device is Tesla k80. Major and minor revision number is printed after it. Global memory is checked with deviceProp.totalGlobalMem function in bytes same constant memory and shared memory is checked for SM. Multiprocessor cont and threads per block is printed after it.

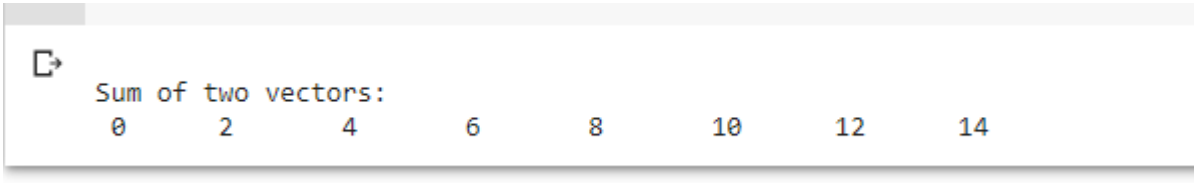
3) Write a CUDA C program to perform the addition of two vectors of arbitrary size (Dynamic Array).

CODE:

```
%%cu
#include <stdint.h>
#include<stdio.h>
#include<unistd.h>

__global__ void vectoradd(int *x,int *y, int *z)
{
  int id=blockIdx.x;
  z[id]=x[id]+y[id];
}

int main(void)
{
    int a[8];
    int b[8];
    int c[8];
    int *d,*e,*f;
    int i;
    for(i=0;i<8;i++)
    {
        a[i]=i;
        b[i]=i;
    }



    cudaMalloc((void **)&d,8*sizeof(int));
    cudaMalloc((void **)&e,8*sizeof(int));
    cudaMalloc((void **)&f,8*sizeof(int));


 cudaMemcpy(d,a,8*sizeof(int),cudaMemcpyHostToDevice);
 cudaMemcpy(e,b,8*sizeof(int),cudaMemcpyHostToDevice);

vectoradd<<<8,1>>>(d,e,f);

cudaMemcpy(c,f,8*sizeof(int),cudaMemcpyDeviceToHost);

printf("\nSum of two vectors:\n ");
    for(i=0;i<8;i++)
    {
        printf("%d\t",c[i]);
    }
```

```
    cudaFree(d);
    cudaFree(e);
    cudaFree(f);

    return 0;
}
```

OUTPUT:



Github link:

https://github.com/shwetaarbune/HPC-LAB7