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

Class: Final Year (Computer Science and Engineering)

Year: 2021-22 **Semester:** 1

Course: High Performance Computing Lab

Practical No. 7

Exam Seat No:2018BTECS00100

1. Exam Seat Number – Prakash Singh

Problem Statement 1: Setup the environment requirements, for execution of CUDA C programs.

Screenshot #:

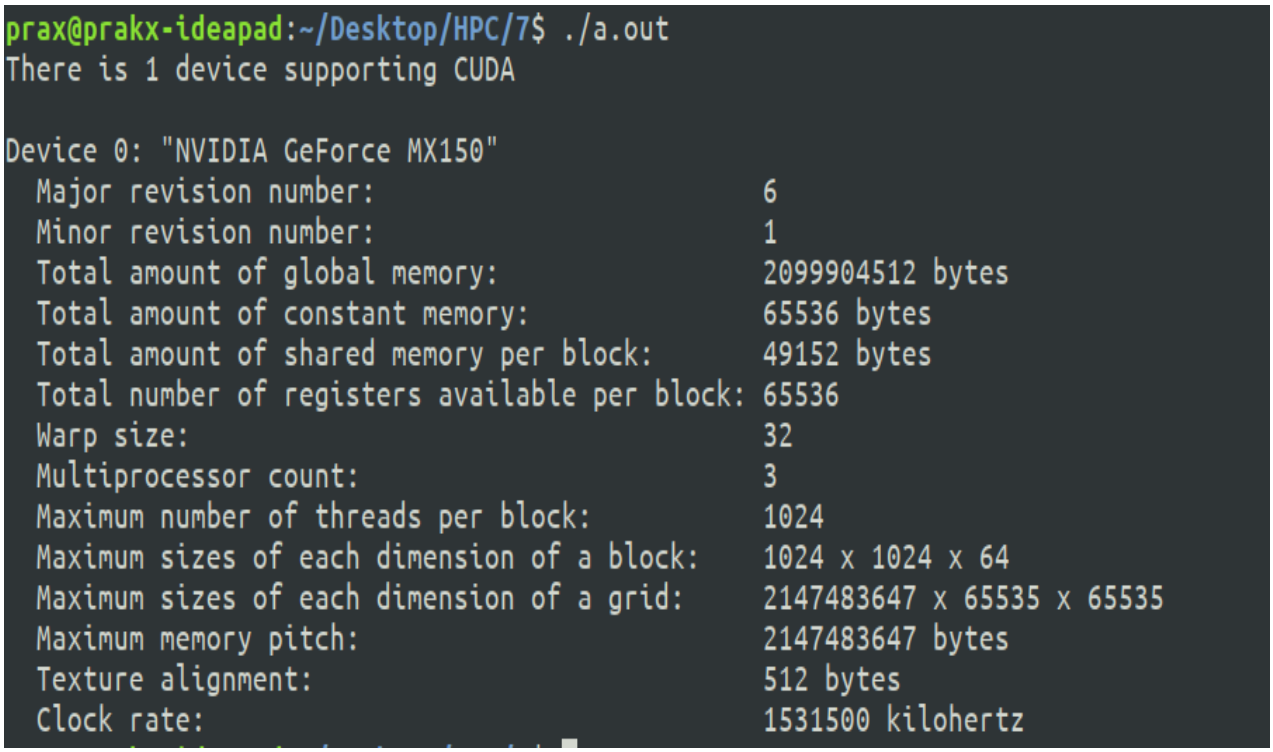
```
prax@prakx-ideapad:~$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
--2021-11-22 17:00:19--  https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 152.199.39.144
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|152.199.39.144|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 190 [application/octet-stream]
Saving to: 'cuda-ubuntu1804.pin'

cuda-ubuntu1804.pin 100%[===================>]     190  --.-KB/s    in 0s

2021-11-22 17:00:19 (2.54 MB/s) - 'cuda-ubuntu1804.pin' saved [190/190]
```

```
prax@prakx-ideapad:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Mon_Sep_13_19:13:29_PDT_2021
Cuda compilation tools, release 11.5, V11.5.50
Build cuda_11.5.r11.5/compiler.30411180_0
prax@prakx-ideapad:~$ S
```

Information #: Installed CUDA 11.5 on ubuntu 18.04 LTS

Problem Statement 2: Execute the attached Program 1, and understand the output.

Screenshot #:



Information #: The program gives device information (NVIDIA GeForce MX150) with global memory of 2GB.

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

Screenshot #:

```c
#include<stdio.h>
#include<cuda.h>

__global__ void arradd(int *x,int *y, int *z)    //kernel definition
{
  int id=blockIdx.x;
/* blockIdx.x gives the respective block id which starts from 0 */
  z[id]=x[id]+y[id];
}

int main()
{
    int a[6];
    int b[6];
    int c[6];
    int *d,*e,*f;
    int i;
    printf("\n Enter six elements of first array\n");
    for(i=0;i<6;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("\n Enter six elements of second array\n");
        for(i=0;i<6;i++)
        {
            scanf("%d",&b[i]);
        }

/* cudaMalloc() allocates memory from Global memory on GPU */
    cudaMalloc((void **)&d,6*sizeof(int));
    cudaMalloc((void **)&e,6*sizeof(int));
    cudaMalloc((void **)&f,6*sizeof(int));

/* cudaMemcpy() copies the contents from destination to source. Here destination is GPU(d,e) and source is CPU(a,b) */
 cudaMemcpy(d,a,6*sizeof(int),cudaMemcpyHostToDevice);
 cudaMemcpy(e,b,6*sizeof(int),cudaMemcpyHostToDevice);

/* call to kernel. Here 6 is number of blocks, 1 is the number of threads per block and d,e,f are the arguments */
arradd<<<6,1>>>(d,e,f);

/* Here we are copying content from GPU(Device) to CPU(Host) */
 cudaMemcpy(c,f,6*sizeof(int),cudaMemcpyDeviceToHost);

printf("\nSum of two arrays:\n ");
    for(i=0;i<6;i++)
    {
        printf("%d\t",c[i]);
    }

/* Free the memory allocated to pointers d,e,f */
    cudaFree(d);
    cudaFree(e);
    cudaFree(f);

    return 0;
}
```

```
prax@prakx-ideapad:~/Desktop/HPC/7$ nvcc vadd.cu
prax@prakx-ideapad:~/Desktop/HPC/7$ ./a.out

 Enter six elements of first array
1 2 2 2 2 2

 Enter six elements of second array
1 2 2 2 2 2

Sum of two arrays:
 2      4       4       4       4       4       prax@prakx-ideapad:~/Desktop/HPC/7$
```

Github Link:https://github.com/prakx1/HPC-LAB/tree/master/7

Note: (Remove this part)

1. Upload only .pdf file on WCE Moodle.
2. Rename .pdf file with ExamSeatNumber_P#
3. Submit Document on moodle and code on Github in public repository.