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

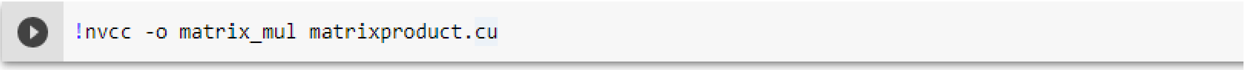
Practical No. 8

Study and Implementation of Grids and Blocks of 2 and 3 Dimensions in CUDA C, GPU Memory and Profiling

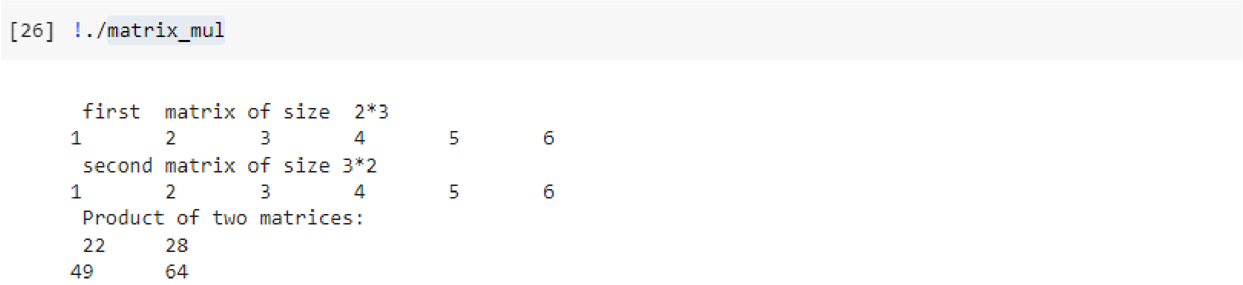
Name : SHWETA NANDKUMAR ARBUNE

PRN:2019BTECS00205

1. Write a CUDA C program to perform the simple matrix-matrix multiplication. Perform code optimization and profiling of existing CUDA C program. (Attach Snapshot of execution before optimization and after optimization)

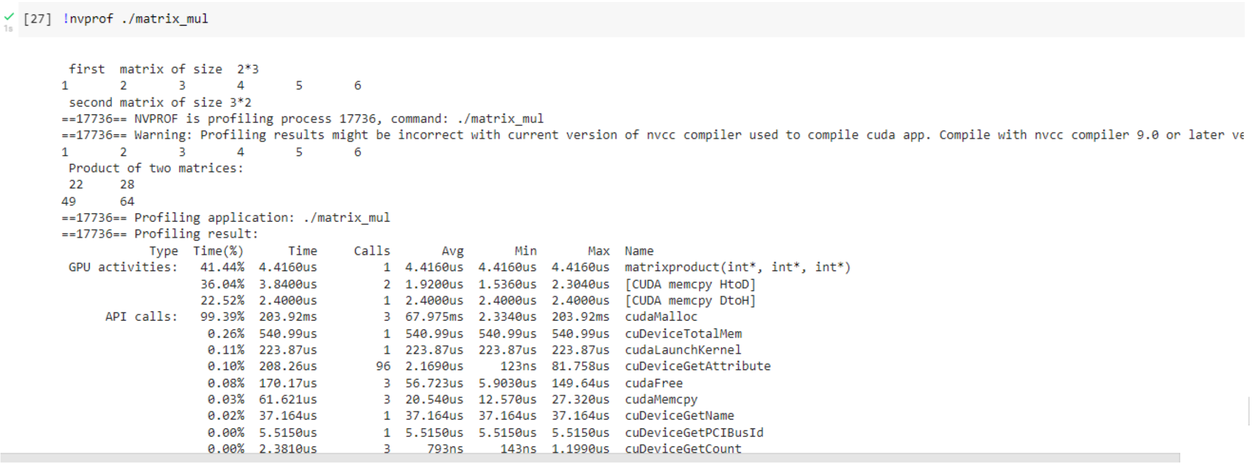


```
!nvcc -o matrix_mul matrixproduct.cu
```



```
[26] !./matrix_mul

 first  matrix of size  2*3
1       2       3       4       5       6
 second matrix of size 3*2
1       2       3       4       5       6
 Product of two matrices:
 22     28
49      64
```

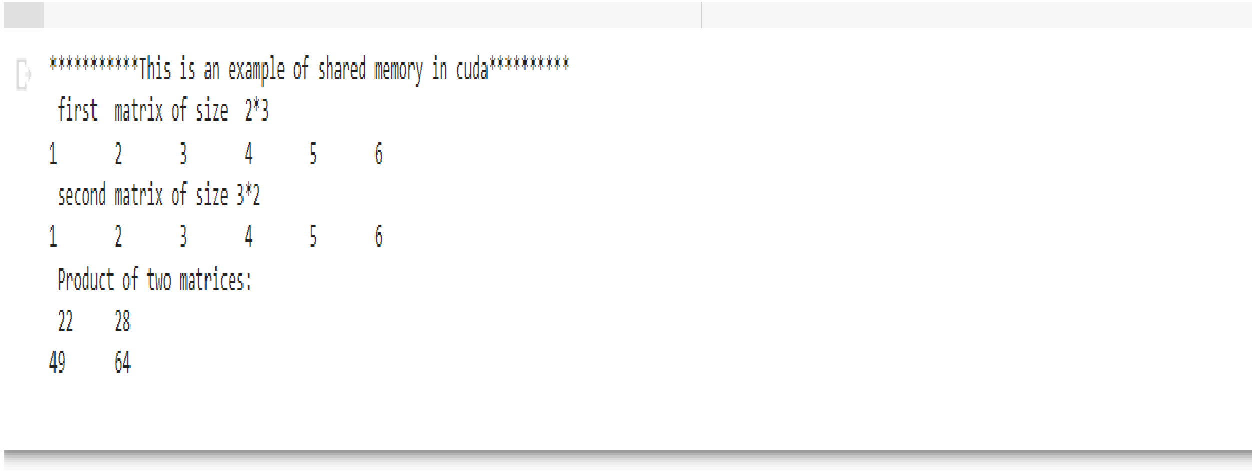


```
[27] !nvprof ./matrix_mul

 first  matrix of size  2*3
1       2       3       4       5       6
 second matrix of size 3*2
==17736== NVPROF is profiling process 17736, command: ./matrix_mul
==17736== Warning: Profiling results might be incorrect with current version of nvcc compiler used to compile cuda app. Compile with nvcc compiler 9.0 or later ve
1       2       3       4       5       6
 Product of two matrices:
 22     28
49      64
==17736== Profiling application: ./matrix_mul
==17736== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   41.44%  4.4160us         1  4.4160us  4.4160us  4.4160us  matrixproduct(int*, int*, int*)
                   36.04%  3.8400us         2  1.9200us  1.5360us  2.3040us  [CUDA memcpy HtoD]
                   22.52%  2.4000us         1  2.4000us  2.4000us  2.4000us  [CUDA memcpy DtoH]
      API calls:   99.39%  203.92ms         3  67.975ms  2.3340us  203.92ms  cudaMalloc
                    0.26%  540.99us         1  540.99us  540.99us  540.99us  cuDeviceTotalMem
                    0.11%  223.87us         1  223.87us  223.87us  223.87us  cudaLaunchKernel
                    0.10%  208.26us        96  2.1690us     123ns  81.758us  cuDeviceGetAttribute
                    0.08%  170.17us         3  56.723us  5.9030us  149.64us  cudaFree
                    0.03%  61.621us         3  20.540us  12.570us  27.320us  cudaMemcpy
                    0.02%  37.164us         1  37.164us  37.164us  37.164us  cuDeviceGetName
                    0.00%  5.5150us         1  5.5150us  5.5150us  5.5150us  cuDeviceGetPCIBusId
                    0.00%  2.3810us         3     793ns     143ns  1.1990us  cuDeviceGetCount
```

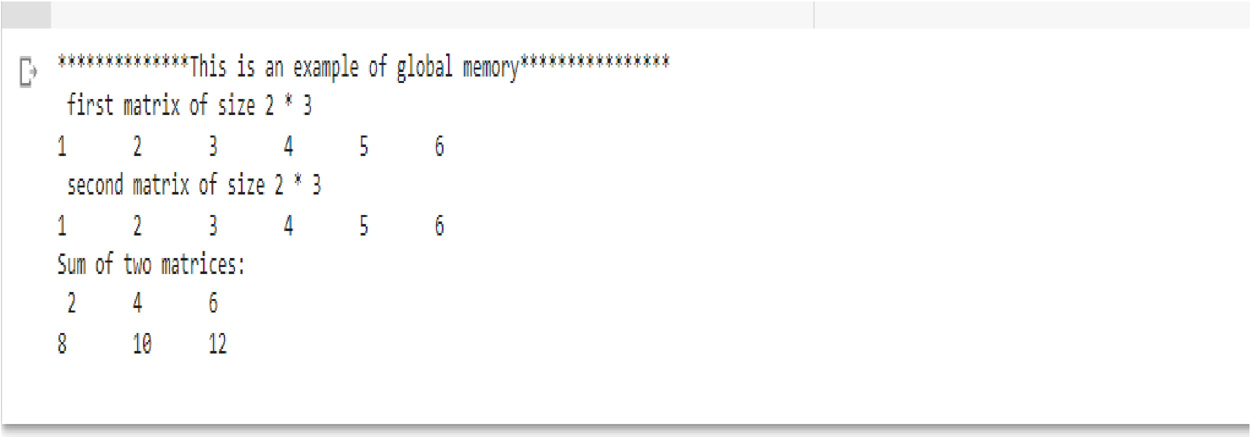
2. Write a CUDA C program to demonstrate the use of different GPU memories.

- Use of shared memory.



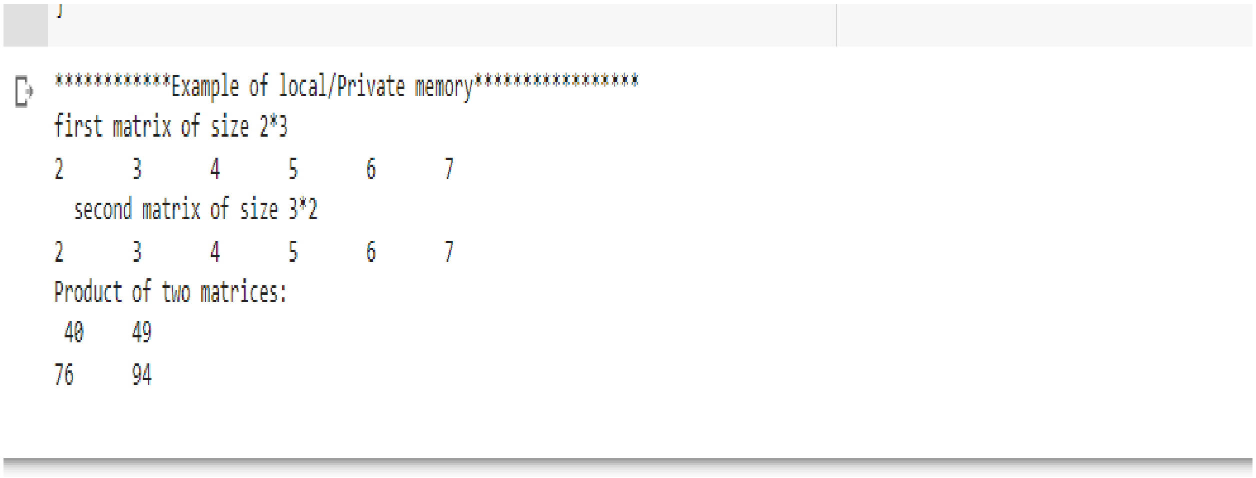
```
***********This is an example of shared memory in cuda**********
 first  matrix of size  2*3
1      2      3      4      5      6
 second matrix of size 3*2
1      2      3      4      5      6
 Product of two matrices:
 22    28
49     64
```

- Use of global memory.



```
**************This is an example of global memory****************
 first matrix of size 2 * 3
1       2       3       4       5       6
 second matrix of size 2 * 3
1       2       3       4       5       6
Sum of two matrices:
 2      4       6
8       10      12
```

- Use of private memory.



```
}
```

```
************Example of local/Private memory*****************
first matrix of size 2*3
2       3       4       5       6       7
  second matrix of size 3*2
2       3       4       5       6       7
Product of two matrices:
 40     49
76      94
```

GitHub profile link: https://github.com/shwetaarbune/HPC-LAB8