Practical 4 :

CUDA Multiplication of Two Matrices :

```
#include <cuda_runtime.h>
#include <iostream>

__global__ void matmul(int* A, int* B, int* C, int N) {
   int Row = blockIdx.y * blockDim.y + threadIdx.y;
   int Col = blockIdx.x * blockDim.x + threadIdx.x;
   if (Row < N && Col < N) {
      int Pvalue = 0;
      for (int k = 0; k < N; k++) {
         Pvalue += A[Row * N + k] * B[k * N + Col];
      }
      C[Row * N + Col] = Pvalue;
   }
}

int main() {
   int N = 512;
   int size = N * N * sizeof(int);
   int* A, * B, * C;
   int* dev_A, * dev_B, * dev_C;
   cudaMallocHost(&A, size);
   cudaMallocHost(&B, size);
   cudaMallocHost(&C, size);
   cudaMalloc(&dev_A, size);
   cudaMalloc(&dev_B, size);
   cudaMalloc(&dev_C, size);

   // Initialize matrices A and B
   for (int i = 0; i < N; i++) {
      for (int j = 0; j < N; j++) {
         A[i * N + j] = i * N + j;
         B[i * N + j] = j * N + i;
      }
   }

   cudaMemcpy(dev_A, A, size, cudaMemcpyHostToDevice);
   cudaMemcpy(dev_B, B, size, cudaMemcpyHostToDevice);

   dim3 dimBlock(16, 16);
   dim3 dimGrid((N + dimBlock.x - 1) / dimBlock.x, (N + dimBlock.y - 1) / dimBlock.y);

   matmul<<<dimGrid, dimBlock>>>(dev_A, dev_B, dev_C, N);
```

```
    cudaMemcpy(C, dev_C, size, cudaMemcpyDeviceToHost);

    // Print the result (first 10x10 elements)
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            std::cout << C[i * N + j] << " ";
        }
        std::cout << std::endl;
    }

    cudaFree(dev_A);
    cudaFree(dev_B);
    cudaFree(dev_C);
    cudaFreeHost(A);
    cudaFreeHost(B);
    cudaFreeHost(C);

    return 0;
}
```

OUTPUT :

207762 209056 210350 211644 212938 214232 215526 216820 218114 219408

523060 526424 529788 533152 536516 539880 543244 546608 549972 553336

838358 843792 849226 854660 860094 865528 870962 876396 881830 887264

1151656 1163160 1174664 1186168 1197672 1209176 1220680 1232184 1243688 1255192

1464960 1482536 1500112 1517688 1535264 1552840 1570416 1587992 1605568 1623144

1778262 1801872 1825482 1849092 1872702 1896312 1919922 1943532 1967142 1990752

2091860 2120592 2149324 2178056 2206788 2235520 2264252 2292984 2321716 2350448

2405056 2438912 2472768 2506624 2540480 2574336 2608192 2642048 2675904 2709760

2718254 2757232 2796210 2835188 2874166 2913144 2952122 2991100 3030078 3069056

3029248 3073360 3117472 3161584 3205696 3249808 3293920 3338032 3382144 3426256