

```

//Write a cuda program for
//1.Addition of two large vectors

#include <iostream>
#include <cuda.h>

__global__ void addVectors(int *A, int *B, int *C, int N) {
    int idx = threadIdx.x + blockIdx.x * blockDim.x;
    if (idx < N) C[idx] = A[idx] + B[idx];
}

int main() {
    int N = 1000000;
    int *A, *B, *C, *d_A, *d_B, *d_C;
    A = new int[N]; B = new int[N]; C = new int[N];

    for (int i = 0; i < N; i++) { A[i] = i; B[i] = i; }

    cudaMalloc(&d_A, N * sizeof(int));
    cudaMalloc(&d_B, N * sizeof(int));
    cudaMalloc(&d_C, N * sizeof(int));

    cudaMemcpy(d_A, A, N * sizeof(int), cudaMemcpyHostToDevice);
    cudaMemcpy(d_B, B, N * sizeof(int), cudaMemcpyHostToDevice);

    addVectors<<<(N + 255) / 256, 256>>>(d_A, d_B, d_C, N);

    cudaMemcpy(C, d_C, N * sizeof(int), cudaMemcpyDeviceToHost);

    std::cout << "C[0] = " << C[0] << ", C[1] = " << C[1] << std::endl; //
Example output
    delete[] A; delete[] B; delete[] C;
    cudaFree(d_A); cudaFree(d_B); cudaFree(d_C);
    return 0;
}

```