#include <iostream>

#include <cuda\_runtime.h>

#define N 1000000 // Vector size

\_\_global\_\_ void vectorAdd(const float\* a, const float\* b, float\* c) {

int i = blockIdx.x \* blockDim.x + threadIdx.x;

if (i < N) {

c[i] = a[i] + b[i];

}

}

int main() {

float\* a = new float[N];

float\* b = new float[N];

float\* c = new float[N];

for (int i = 0; i < N; i++) {

a[i] = i;

b[i] = 2 \* i;

}

float\* dev\_a, \* dev\_b, \* dev\_c;

cudaMalloc(&dev\_a, N \* sizeof(float));

cudaMalloc(&dev\_b, N \* sizeof(float));

cudaMalloc(&dev\_c, N \* sizeof(float));

cudaMemcpy(dev\_a, a, N \* sizeof(float), cudaMemcpyHostToDevice);

cudaMemcpy(dev\_b, b, N \* sizeof(float), cudaMemcpyHostToDevice);

int threadsPerBlock = 256;

int blocksPerGrid = (N + threadsPerBlock - 1) / threadsPerBlock;

vectorAdd << <blocksPerGrid, threadsPerBlock >> > (dev\_a, dev\_b, dev\_c);

cudaMemcpy(c, dev\_c, N \* sizeof(float), cudaMemcpyDeviceToHost);

cudaFree(dev\_a);

cudaFree(dev\_b);

cudaFree(dev\_c);

for (int i = 0; i < N; i++) {

if (c[i] != a[i] + b[i]) {

std::cout << "Error: mismatch at index " << i << std::endl;

break;

}

}

delete[] a;

delete[] b;

delete[] c;

return 0;

}