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

#define N 1000000 // Define the size of the vectors

// CUDA kernel function for vector addition
__global__ void vectorAdd(float *A, float *B, float *C) {
    int i = blockIdx.x * blockDim.x + threadIdx.x; // Calculate the index
    if (i < N) {
        C[i] = A[i] + B[i]; // Add the vectors element-wise
    }
}

int main() {
    // Allocate memory on the host
    float *h_A = (float*) malloc(N * sizeof(float));
    float *h_B = (float*) malloc(N * sizeof(float));
    float *h_C = (float*) malloc(N * sizeof(float));

    // Initialize vectors on the host
    for (int i = 0; i < N; i++) {
        h_A[i] = i * 1.0f;
        h_B[i] = i * 2.0f;
    }

    // Allocate memory on the device
    float *d_A, *d_B, *d_C;
    cudaMalloc((void**)&d_A, N * sizeof(float));
    cudaMalloc((void**)&d_B, N * sizeof(float));
    cudaMalloc((void**)&d_C, N * sizeof(float));

    // Transfer data from host to device
    cudaMemcpy(d_A, h_A, N * sizeof(float), cudaMemcpyHostToDevice);
    cudaMemcpy(d_B, h_B, N * sizeof(float), cudaMemcpyHostToDevice);

    // Configure the kernel launch parameters
    int blockSize = 256; // Number of threads per block
    int numBlocks = (N + blockSize - 1) / blockSize; // Number of blocks

    // Launch the kernel
    vectorAdd<<<numBlocks, blockSize>>>(d_A, d_B, d_C);

    // Transfer data from device to host
    cudaMemcpy(h_C, d_C, N * sizeof(float), cudaMemcpyDeviceToHost);

    // Verify the results
    for (int i = 0; i < 10; i++) {
        printf("%f + %f = %f\n", h_A[i], h_B[i], h_C[i]);
    }

    // Free device memory
    cudaFree(d_A);
    cudaFree(d_B);
    cudaFree(d_C);

    // Free host memory
    free(h_A);
    free(h_B);
    free(h_C);
    return 0;
}


#OUTPUT:

0.000000 + 0.000000 = 0.000000
1.000000 + 2.000000 = 3.000000
2.000000 + 4.000000 = 6.000000
3.000000 + 6.000000 = 9.000000
4.000000 + 8.000000 = 12.000000
5.000000 + 10.000000 = 15.000000
6.000000 + 12.000000 = 18.000000
7.000000 + 14.000000 = 21.000000
8.000000 + 16.000000 = 24.000000
9.000000 + 18.000000 = 27.000000
```