Assignment 4A - Vector Addition

```
!nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

```
code = """

#include <stdio.h>
#include <stdlib.h>

#define N 1000000

__global__ void add(int *a, int *b, int *c) {
    int tid = blockIdx.x * blockDim.x + threadIdx.x;
    if (tid < N) {
        c[tid] = a[tid] + b[tid];
    }
}

int main() {
    int *a, *b, *c;
    int *d_a, *d_b, *d_c;
    int size = N * sizeof(int);

    // Allocate memory on host
    a = (int*)malloc(size);
    b = (int*)malloc(size);
    c = (int*)malloc(size);

    // Initialize arrays
    for (int i = 0; i < N; i++) {
        a[i] = i;
        b[i] = i * 2;
    }

    // Allocate memory on device
    cudaMalloc(&d_a, size);
    cudaMalloc(&d_b, size);
    cudaMalloc(&d_c, size);

    // Copy data from host to device
    cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
    cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);

    // Launch kernel with 1 million threads
    add<<<(N + 255) / 256, 256>>>(d_a, d_b, d_c);

    // Copy result from device to host
    cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);

    // Print first and last elements of result
    printf("c[0]=%d, c[%d] = %d",c[0],N-1,c[N-1]);

    // Free memory
    free(a);
    free(b);
    free(c);
    cudaFree(d_a);
    cudaFree(d_b);
    cudaFree(d_c);

    return 0;
}
"""
```

```
text_file = open("assign4.cu","w")
text_file.write(code)
text_file.close()
```

```
!nvcc assign4.cu
```

```
!./a.out
```

```
c[0]=0, c[999999] = 2999997
```

```
!nvprof ./a.out
```

```
==1065== NVPROF is profiling process 1065, command: ./a.out
==1065== Profiling application: ./a.out
==1065== Profiling result:
```

```
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   51.91%  1.9264ms         1  1.9264ms  1.9264ms  1.9264ms  [CUDA memcpy DtoH]
                   46.81%  1.7372ms         2  868.62us  813.34us  923.90us  [CUDA memcpy HtoD]
                    1.28%  47.455us         1  47.455us  47.455us  47.455us  add(int*, int*, int*)
      API calls:   97.16%  268.79ms         3  89.598ms  100.42us  268.58ms  cudaMalloc
                    2.08%  5.7679ms         3  1.9226ms  1.0787ms  3.5174ms  cudaMemcpy
                    0.44%  1.2053ms         1  1.2053ms  1.2053ms  1.2053ms  cuDeviceGetPCIBusId
                    0.24%  676.53us         3  225.51us  205.02us  237.71us  cudaFree
                    0.05%  143.16us       101  1.4170us     133ns  73.062us  cuDeviceGetAttribute
                    0.01%  37.564us         1  37.564us  37.564us  37.564us  cudaLaunchKernel
                    0.01%  25.957us         1  25.957us  25.957us  25.957us  cuDeviceGetName
                    0.00%  1.6580us         3     552ns     218ns  1.2050us  cuDeviceGetCount
                    0.00%  1.0080us         2     504ns     192ns     816ns  cuDeviceGet
                    0.00%     505ns         1     505ns     505ns     505ns  cuModuleGetLoadingMode
                    0.00%     447ns         1     447ns     447ns     447ns  cuDeviceTotalMem
                    0.00%     280ns         1     280ns     280ns     280ns  cuDeviceGetUuid
c[0]=0, c[999999] = 2999997
```

✓ 1s completed at 3:53 PM