Q4

We write vector addition using both CUDA and OpenACC. Here is the result of both on two different GPUs: v100-pcie and v100-sxm2

OpenACC code:

```
    start = CLOCK();

#pragma acc kernels
    for (i = 0; i < n; i++)
        h_c[i] = h_a[i] + h_b[i];

    // Sum up vector c and print result divided by n, this should equal 1 within error
    double sum = 0;
    for (i = 0; i < n; i++)
        sum += h_c[i];

    finish = CLOCK();
```

GPU	CUDA	OpenACC
v100-pcie	508.02 ms	4832.10 ms
v100-sxm2	85.89 ms	2057.35 ms

```
[zhang.yam@c2205 Question4]$ ./Q4CUDA
final result: 1.000000
Time for the loop = 508.02 milliseconds
[zhang.yam@c2205 Question4]$ ./Q4ACC
final result: 1.000000
Time for the loop = 4832.10 milliseconds
```

```
[zhang.yam@d1017 Question4]$ ./Q4CUDA
final result: 1.000000
Time for the loop = 85.89 milliseconds
[zhang.yam@d1017 Question4]$ ./Q4ACC
final result: 1.000000
Time for the loop = 2057.35 milliseconds
```

The Running time of OpenACC is lower than CUDA, however, the code of OpenACC is much easier than CUDA. We do not need to care about the grid size and block size. And the comparison between two GPUs v100-pcie and v100-sxm2 shows that v100-sxm2 is way faster than v100-pcie.