Report

任务一

运行 `run.sh` 得到如下输出：

```
+ set -e
+ make -j
icpc aplusb-baseline.cpp -c -o aplusb-baseline.o -g
icpc aplusb-autosimd.cpp -c -o aplusb-autosimd.o -g -xhost
icpc aplusb-intrinsic.cpp -c -o aplusb-intrinsic.o -g -xhost
icpc -std=c++17 aplusb.cpp aplusb-baseline.o aplusb-autosimd.o aplusb-intrinsic
+ srun ./aplusb
baseline: 4431 us
auto simd: 498 us
Wrong answer at 0: c1 = 1.224684, c3 = 0.000000
srun: error: conv2: task 0: Exited with exit code 1
```

可以看出运行时间从原来的 `4431 us` 优化到了 `498 us` 。带来的优化效果非常明显。

任务二

在完成代码后，运行 `run.sh` 得到如下输出：

```
+ set -e
+ make -j
make: Nothing to be done for 'all'.
+ srun ./aplusb
srun: job 605772 queued and waiting for resources
srun: job 605772 has been allocated resources
baseline: 4436 us
auto simd: 499 us
intrinsic: 522 us
```

基于 intrinsic 的手动向量化获得了和编译器自动向量化相近的性能。

`a_plus_b_intrinsic` 函数代码如下：

```
void a_plus_b_intrinsic(float* a, float* b, float* c, int n) {
    for (int i = 0; i < n; i += 8)
    {
        __m256 m1 = _mm256_load_ps(a + i);
        __m256 m2 = _mm256_load_ps(b + i);
        __m256 m = _mm256_add_ps(m1, m2);
        _mm256_store_ps(c + i, m);
    }
}
```