

```
1 using std::vector;
2 void vecAddLong(float *A, float *B, float *C, int n,
3                 accelerator_view acc)
4 {
5     int block = (acc.accelerator.dedicated_memory * 1024)
6                 /(3*sizeof(float));
7     vector<completion_future> results;
8     for(int i = 0; i < n; i += block) {
9         int m = min(n-i,block);
10        array_view<const float,1> AV(m,A+i), BV(m,B+i);
11        array_view<float,1> CV(m,C+i);
12        CV.discard_data();
13        parallel_for_each(acc, CV.extent, [=](index<1> idx) restrict(amp)
14        {
15            CV[idx] = AV[idx] + BV[idx];
16        });
17        results.push_back(CV.synchronize_async());
18    }
19    std::for_each(results.begin(), results.end(),
20                 [] (completion_future f) { f.get(); });
21 }
```