

```

1  __global__ void ker() {
2      __shared__ int A[10];
3      int tid = threadIdx.x;
4      __check(A, tid);    // Cdom
5      if (tid % 32 > 28) {
6          __check(A, tid); // Csub
7      }
8  }

```

(a) Recurring check.

```

1  __global__ void ker(int* A, int N) {
2      int l = tIdx.x, s = bDim.x;
3      for (int i=l; i<N; i+=s) {
4          __check(A, i+2);    // Cmax
5          // __check(A, i+1);
6          __check(A, i+0);    // Cmin
7      }
8  }

```

(b) Neighboring check.

```

1  __global__ void ker(int* A, int N) {
2      int l=tIdx.x, s=bDim.x;
3      __check(A, 0);          // Cmin
4      __check(A, N - 1);     // Cmax
5      for (int i=l; i<N; i+=s) {
6          __check(A, i);
7      }
8  }

```

(c) Loop-inductive check.