```
   __kernel void matrixMul( __global float* C, __global float* A,
         __global float* B, __local float* AS, __local float* BS,
         int uiWA, int uiWB )
   {
1    int aBegin = uiWA * BLOCK_SIZE * Gid.y;
2    int aEnd   = aBegin + uiWA - 1;
3    int aStep  = BLOCK_SIZE;
4    int bBegin = BLOCK_SIZE * Gid.x;
5    int bStep  = BLOCK_SIZE * uiWB;

6    float Csub = 0.0f;
7    for (int a = aBegin, b = bBegin; a <= aEnd;
          a += aStep, b += bStep)
     {
8        AS[Lid.x + Lid.y * BLOCK_SIZE] = A[a + uiWA * Lid.y + Lid.x];
9        BS[Lid.x + Lid.y * BLOCK_SIZE] = B[b + uiWB * Lid.y + Lid.x];
10       barrier(CLK_LOCAL_MEM_FENCE);
11       for (int k = 0; k < BLOCK_SIZE; ++k)
12         Csub += AS[k + Lid.y*BLOCK_SIZE] * BS[Lid.x + k*BLOCK_SIZE];
13       barrier(CLK_LOCAL_MEM_FENCE);
     }
14   C[(Gid.y*GROUP_SIZE_Y+Lid.y)*GLOBAL_SIZE_X
                                  + (Gid.x*GROUP_SIZE_X+Lid.x)] = Csub;
   }
```