```
#include "matrixmult.h"

// Matrix multiplication kernel called by MatMul()
__global__ void MatMulKernel(Matrix A, Matrix B, Matrix C)
{
   /* Each thread computes one element of C
      by accumulating results into Cvalue
   */
   float Cvalue = 0;
   int k;
   int row = blockIdx.y * blockDim.y + threadIdx.y;
   int col = blockIdx.x * blockDim.x + threadIdx.x;
   for ( k = 0; k < A.width; k++)
      Cvalue += A.elements[row * A.width + k] *
                B.elements[k * B.width + col];

   C.elements[row * C.width + col] = Cvalue;
}
```