1. Explain how you increased memory bandwidth over the naive GPU version that only uses global memory.
   Each time read the naive GPU version will load 12 times global memory from global memory per thread, if we put it into shared memory, each block will have at most (blocksize + 2) * (blockszie + 2) read from global memory.
2. Explain why you selected 1D or 2D structure for your blocks.
   2D structure, because it is like a the mask part int the 2D Convolution, each thread will use the nearest 8 data in 2D structure, it is more convenient to use 2D structure.
3. Describe your implementation for fetching pixels at the boundaries of a block that are needed for computing the filter for interior pixels.
   Set all the boundary output to 0, for the inner one, If the threadIdx.y is 0 or 1, the thread will load the previous and next block size's data.

```
s_data[threadIdx.x][threadIdx.y * (TILESIZE + 1)
s_data[threadIdx.y * (TILESIZE + 1)][threadIdx.x + 1]
```

The top one for the outer x = 0 or 17, below one for y = 0 or 17, and need to load extra 4 points, the index of my implement have some problems always fail.