Matrix addition is very similar to vector addition: you just add the corresponding entries of two two-dimensional arrays. Serial pseudo-code might look something like this:

```c
float A[m][n], B[m][n], C[m][n];
...
// Initialize A and B
...
// Add A and B
for (int i = 0; i < m; i++)
    for (int j = 0; j < n; j++)
        C[i][j] = A[i][j] + B[i][j];
```

Of course, in CS 220 we use one-dimensional arrays, and the assignment statement would look something like

```c
C[i*n + j] = A[i*n + j] + B[i*n + j];
```

For this assignment you should write a CUDA program that implements matrix addition. The host should get \(m\) and \(n\) from the command line, allocate storage for \(A, B,\) and \(C\) on both the host and device, and read \(A\) and \(B\) from `stdin`. After \(A\) and \(B\) have been initialized on the host and the device, the host should call a kernel that adds \(A\) and \(B\). Each thread block should compute a row of \(C\), and each thread should compute a single element of \(C\). When the kernel completes, your code should synchronize the host and the device, copy \(C\) to the host and print it out.

The `vec_add` program we went over in class may be useful.