Homework 3

Problem 1. Assume a cache line size of 4 elements. Identify the different kinds of reuse and how many memory accesses there are in the following example, assuming (a) row-major order, (b) column-major order. Use the inner loop memory cost calculation from slides 11-13 of Lecture 15 to estimate memory accesses.

```
for (i = 0; i<n; i++)
  for (j = 0; j<m; j++)
      A[i][j] = B[i][j] + B[j][i] + C[i]
```

Homework 3, cont.

Problem 2. What code would be generated to tile the following loop nest for reuse in cache, assuming row-major order and two levels of tiling (Note: the loop order may need to be modified). Prove that tiling is safe.

```
for (i = 0; i<n; i++)
    for (j = 0; j<m; j++)
      for (k=0; k<l;k++)
          A[i][k] = A[i][k] + B[k][j]*C[k][k]
```

Homework 3

Problem 3: VTUNE:

Consider the jacobi code in jacobi.c on the website. Here is the main computation:

```
// play around with this loop nest
   for (i=1; i<width-1; i++)
     for (j=1; j<height-1; j++)
       A[i][j] = (B[i+1][j] + B[i-1][j] + B[i][j+1] + B[i][j-1])/4;
```

(a) Run this code under VTUNE. Indicate event-based sampling, and collect the following events. (CPU_CLK_UNHALTED, MEM_LOAD_RETIRED.L1D_MISS, MEM_LOAD_RETIRED.L2_MISS)

(b) Now attempt to tile the innermost loop and repeat. Do you see an impact on cache misses and cycles.

(c) Tile the other loop. Now what happens.