Lab 4 Report
Jillian Empey and Levi Thompson

Unroll: 0

	-O0	-O1	-O2	-O3
cycles:	1366143794	1365659391	1371795623	1372326810
instructions:	1067200076	1067021816	1067023709	1067213612
IPC:	0.78	0.78	0.78	0.78
CPI:	1.282	1.282	1.282	1.282
misses:	6672670	6681616	6700521	6677360
% misses:	6.13%	6.14%	6.16%	6.13%
seconds:	1.730834576	1.782069633	1.788375825	1.743144251

Unroll: 2

	-O0	-O1	-O2	-O3
cycles:	1365403123	1367189699	1364039468	1365434522
instructions	1056569566	1056562473	1056589934	1056745089
IPC:	0.77	0.77	0.77	0.77
CPI:	1.299	1.299	1.299	1.299
misses:	6693097	6719570	6697532	6695094
% misses:	6.80%	6.83%	6.81%	6.81%
seconds:	1.730017232	1.740039379	1.777740341	1.728580520

Unroll: 4

	-O0	-O1	-O2	-O3
cycles:	1365914053	1365807573	1365450647	1372124988
instructions:	1051322229	1051323801	1051489893	1051325384
IPC:	0.77	0.77	0.77	0.77
CPI:	1.299	1.299	1.299	1.299
misses:	6702492	6659038	6694175	6652378
% misses:	7.20%	7.15%	7.19%	7.14%
seconds:	1.779401105	1.728635738	1.779484436	1.739811312

Unroll: 8

	-O0	-O1	-O2	-O3
cycles:	1371759546	1365246193	1361196344	1372721435
instructions:	1048862320	1048698946	1048805854	1048704540
IPC:	0.76	0.77	0.77	0.76
CPI:	1.316	1.299	1.299	1.316
misses:	6687757	6690476	6751751	6718221
% misses:	7.39%	7.39%	7.46%	7.42%
seconds:	1.739183958	1.726746568	1.785130311	1.740688991

```
Each sample counts as 0.01 seconds.
  %   cumulative   self
 time   seconds   seconds
 93.33      0.28     0.28     cols(** majority of matadd here)
  3.33      0.29     0.01     AllocateMatrix(unsigned int, unsigned int, int)
  3.33      0.30     0.01     PrintMat(int**, unsigned int, unsigned int)
```

```
Calculate E Using Amdahl's Law
T(impr) = T(affected)/Improvement Factor + T(unaffected)
   Improvement Factor From No Unrolling to 2 Unrolling (-O1):
      Ti = 1.740039379
      Ta = (.9333)(1.782069633)
      Tu = 1.782069633-Ta
      IF = Ta/(Ti-Tu)
         = 1.66/(1.74-.1189) = 1.024x Faster
   Improvement Factor From No Unrolling to 4 Unrolling:
      Ti = 1.729
      Ta = 1.663
      Tu = .1189
      IF = Ta/(Ti-Tu)
         = 1.663/(1.729-.1198) = 1.033x Faster
   Improvement Factor From No Unrolling to 8 Unrolling:
      Ti = 1.727
      Ta = 1.663
      Tu = .1189
      IF = Ta/(Ti-Tu)
         = 1.663/(1.727-.1198) = 1.035x Faster
Performance Equation
   CPU Time = (# Instructions/program)(CPI)(seconds/clock cycle)

   For -O1:
       Texp = (1067021816 instructions/program)*(1.282 cycles/instr)
               *(1.782069633 seconds/1365659391)
            = 1.784 seconds ~= execution time
```

Using the performance equation, the expected CPU time almost exactly equaled the measured expectation time (with a percent error of .16%!)

There was not a huge improvement observed which is kind of expected on these machines considering they aren't incredibly advanced. However, even though there wasn't much of a speedup in execution time, there were tens of millions less instructions in more unrolled executions which could be significant on a better machine.

The compiler optimizes code by further unrolling it in order to decrease the number of breaks (which take time and extra instructions).