CS 6560 Parallel Computer Architecture

Assignment 2: Due on 16/02/2016 @ 2359hrs

1. Consider the following programming to compute the sum of two matrices A and B

```
#include <stdio.h>
int main()
{
      int N = 100000;
      for (int i = 0; i < N; i++) {
            X[i] = A[i] + B[i];
      }
      return 0;
}
```

   a) Parallelize this program with pthreads that uses 10 threads; partition the array such that the load is balances across all threads. Include your C file, and command used to compile the program (using gcc) in your solution.
   b) Parallelize this program with OpenMP by using 10 threads with the same goal as above. Include your C code and commands use to compile the program (using gcc).
   c) Assuming that each integer is 8B, and your program is running a multi-processor system that has a single level of cache that is 32KB. How many threads would you spawn in your software to reduce the effect of cache misses, when ignoring the # hardware threads?
   d) In c), if you're told that the HW has 8 threads, would you change your implementation? If so, how?

2. What would be the output of the following program if the environment variable OMP_NUM_THREADS is set to (a) 1, (b) 2, (c) 3

```
#include <stdio.h>
#include <omp.h>
#include <stdlib.h>
int main (int argc, int *argv[])
{
      int X = 2;
      #pragma omp parallel
      {
            printf("Thread: %d, the value of the variable x
                        is %d\n", omp_get_thread_num(), X);
            X = 0;
      }
      return 0;
}
```

   What changes, if any, would you suggest to this program?

3. Describe Intel's quick-sync layer, and the various functionalities that it contains. How can a programmer take advantage of the Quick-sync layer? Name 4 software tools that take advantage of this layer today (in under 100 words).