Homework 2

Problem 1 (15%): Compute Cache performance is a factor of several parameters. For each of these, describe the issues that arise if their value is either too small or too large:

a. Cache size
b. Line size
c. Associativity

Problem 2 (50%): Consider the matrix_add function shown below:

```
int matrix_add(int a[128][128], int b[128][128], int c[128][128])
{
   int i, j;
   for(i = 0; i < 128; i++)
    for(j = 0;j < 128; j++)
       c[i][j] = a[i][j] + b[i][j];
       return 0;
}
```

In each iteration, the compiled code will load a[i][j] first, and then load b[i][j]. After performing the addition of a[i][j] and b[i][j], the result will be stored to c[i][j]. The processor has a 64KB, 2-way, 64Byte-block L1 data cache, and the cache uses LRU policy once a set if full. The L1 data cache is write-back and write-allocate. If the addresses of array a, b, c are 0x10000, 0x20000, 0x30000 and the cache is currently empty, please answer the following questions:

a. What is the L1 D-cache miss rate of the matrix_add function? How many misses are contributed by compulsory miss? How many misses are conflict misses?

b. If the L1 hit time is 1 cycle, and the L1 miss penalty is 20 cycles. What is the average memory access time?

Problem 3 (35%): You are given a cache that has 16 byte blocks, 512 rows, and is 2-way set associative. Integers are 4 bytes. Give the C code for a loop that has a 100% miss rate in this cache but whose hit rate rises to almost 100% if you double the size of the cache.