1. An arithmetic unit is implemented in two ways:
   - As a four-stage arithmetic pipeline $P$ where the four stages, $S_1$, $S_2$, $S_3$, and $S_4$ have combinational delays of 1 unit, 1.5 units, 2.5 unit, and 1 units, respectively. All operands are fed into $S_1$ and the output is provided by $S_4$.
   - As an unpipelined implementation $U$, identical to $P$ except that all registers are removed. For simplicity, assume that all register delays and set-up times are zero.

   (a) What is the best-case speedup of pipeline $P$ over the implementation $U$? Under what conditions will this speedup be achieved?

   (b) The delay of a pipeline stage corresponds to the path with the largest delay. For stage $S_3$, only 20% all operations are "fast" and excite this worst-case delay path, while the remaining 80% are "fast" and have a maximum delay of 1.3 units.

   Suppose we alter the pipeline to a configuration $P'$, which still has four stages as in $P$, but where the "fast" operations in $S_3$ are completed in one cycle, and the "slow" operations in $S_3$ require two cycles. Note that while a "slow" operation is being executed, stage $S_3$ is busy and the pipeline is appropriately stalled until $S_3$ becomes available. Over a large number of operations where the mix of "fast" and "slow" operations follows the 80%/20% distribution, what is the best-case speedup of the pipeline $P'$ over $U$?

   (c) Now, consider the case where the delays of the stages in the original pipeline $P$ are balanced so that each stage now has a delay of 1.5 units, and unlike (b), each stage requires exactly one cycle. Call this pipeline $P''$. What is the best-case speedup of pipeline $P''$ over the implementation $U$? Under what conditions will this speedup be achieved? [1.5 points]

2. A computation requires two types of operations: (i) multiply operations and (ii) memory fetch operations which are inherently serial. On a serial machine, at any snapshot in time, 75% of all operations are multiplications while the rest are memory fetches.

   (a) If we were to parallelize the multiply operations, how many multipliers must operate in parallel in order to yield a speedup of 6 over the serial case?

   (b) What is the best achievable speedup? [1 point]

3. Consider the design of a memory system that has a single level of cache of size 512 Kilobytes with a block size of 64 words, a main memory of size 1 Gigabyte, and a secondary memory. The memory is word-addressable, where each word is 32 bits long. Explain precisely how a main memory address is mapped on to the cache when the cache is:

   i) fully associative
   ii) direct-mapped
   iii) 4-way set associative [1.5 points]