1. You have some protein matching code. It completes in 200 hours on your current machine, and spends 20% of the time doing integer operations.

   (a) How much faster must you make the integer unit to make the code run **10 hours faster**?

   (b) How much faster must you make the integer unit to make the code run **50 hours faster**?

2. You have some graph processing code which takes four days to execute on your current machine. Of that time

   - 20% of the time is spent performing integer operations, and
   - 35% of the time is spent performing I/O operations.

   Which of the following is the better tradeoff?

   (a) A compiler optimization that reduces the number of integer instructions by 25% (assume that each integer operation still takes the same amount of time).

   (b) A hardware optimization that reduces the latency of each I/O operation from $6\mu s$ to $5\mu s$.

3. Recent advances in process technology have quadrupled the number of transistors you can fit on your processor die. Currently, your key customer can use up to 4 processors for 40% of their application.

   You have two choices:

   (a) Increase the number of processors from 1 to 4.

   (b) Increase the number of processors from 1 to 2, but add features to each processor that will allow the application to use 2 processors for 80% of the execution time.

   Which is the best choice?

4. In your application, memory operations currently take 30% of the execution time.

   A new widget called "cache" speeds up 80% of memory operations by a factor of 4.

   A second new widget called "L2 cache" speeds up half of the remaining memory operations by a factor of 2.

   What is the total speedup?