1. **True or False:**

(1) Amdahl's law doesn't apply to parallel computers.

(1) It is possible to design a flawless architecture.

(1) There is such a thing as a typical program.

(1) Multiprocessors are "free".

(1) You can predict cache performance of Program A by analyzing Program B.

(1) An architecture with flaws cannot be successful.

(1) Linear speedups are needed to make multiprocessors cost-effective.

(1) Scalability is almost free.

2. (2) What is the main reason few modern desktop/server class machines have a stack-based instruction set, and no supercomputers do?

3. (2) Why do most high-performance machines use a fixed instrution size? (What is the difficulty that variable length instructions present?)

4. (2) In your own words, what does Amdahl's law say?

5. (4) The book lists many Compiler optimizations. Write down 4 of them.

6. (6) List at least 4 different addressing modes, and describe (in words or pictures) where the operand is stored.

7. (4) List Flynn's 4 different catagories of parallel processors.

8. (30) Assume that the CPI of a processor with a perfect cache is 2.0, the clock cycle time is 1 ns, and there are 1.5 memory references per instruction, and the cache has a block size of 64 bytes. System A (Happy) uses a direct mapped cache, while System B (Sneezy) uses a two-way set associative one. Since a set-associative cache requires an extra multiplexor/selector, the cycle time of Sneezy is 1.25 longer than that of Happy. The cache miss penalty is 75ns for both systems.

a.) Calculate the average memory access time and CPU performance for each processor. Assume the hit time is 1 clock cycle, the miss rate of a direct-mapped 64KB cache is 1.4%, and the miss rate for a 64KB two-way set associative cache is 1.0%. Which system is faster?

b.) Now assume that Sneezy is modified to support Out Of Order exection, and also has it's set-associative cache replaced with a direct-mapped one. The clock cycle time stays the same as it did (1.25 times that of Happy). Assume 30% of the 75ns miss penalty can be hidden by overlapping it with other instructions; that is, the average CPU memory stall time is now 52.5ns. How does the performance of Sneezy now compare to that of Happy?

9. (16) Consider the following description of a computer (Grumpy) and its cache performance:

Block size = 1 word

Memory bus width = 1 word

Miss rate = 3%

Memory accesses per instruction = 1.2

Cache miss penalty = 64 cycles

Average cycles per instruction (ignoring cache misses) = 2

If we change the block size to 2 words, the miss rate falls to 2%, and a 4-word block has a miss rate of 1.2%. What is the improvement in performance if memory is interleaved two ways and four ways, versus doubling the width of memory and the bus? Assume it takes 4 clock cycles to send an address, the access time is 56 clock cyles per word, and it takes 4 clock cycles to send a word of data.

10. (16) Suppose you have an application running on a 32-processor multiprocessor, which requires 400 ns to service a reference to a remote memory (pass a message, essentially). Assume that all references except those involving communication hit in the local memory hierarchy. Processors stall when they issue a remote request, and each processor has a 1GHz clock. If the base IPC (assuming all references hit in the cache) is 2, how much faster is the multiprocessor if there is no communication, vs if .2% of the instructions involve a remote communication reference?

11. (10) Given the following high-level language code fragment:

```
A = B + C;
B = A + C;
D = A - B;
```

Using the Copy Propagate technique, rewrite this code fragment so that the number of computed values is minimized. Note which statements in your transformed code are now doing more work, and which ones are doing less. What does this suggest regarding the technical challenges faced by writers of optimizing compilers?