

---

Please state clearly any assumptions you make in solving the following problems.

---

## 1 Languages and instruction sets

Write a short report comparing at least two computer languages (such as C, Fortran, Python, Java, ...) versus each other and at least two instruction set architectures (such as ARM, x86, SPARC, ...) versus each other. You may benefit from <http://cpudb.stanford.edu/> and <https://www.cpubenchmark.net/>.

(5 points)

---

## 2 Classical papers versus current reality

1. Read the paper “The evolution of RISC technology at IBM” by John Cocke and Victoria Markstein. (1 point)
  2. What are the reasons given in the paper to make all the instructions 32 bits long? (1 point)
  3. What was needed to support decimal arithmetic? (1 point)
  4. What is the difference between the rotate and shift instructions and what are their uses? (1 point)
  5. Evaluate the two instruction set architectures that you compared in problem 1 according to the architectural aims given in the summary section of the paper. (1 point)
- 

## 3 Improving performance

A smart cell phone, used for web surfing, watching videos, and occasional calls, has a processor running at a 400 MHz clock, a 10 million polygon/sec graphics chip and a 200 KBps data connection. When watching a video 50 percent of the execution time is graphics, 20 percent is CPU use and 30 percent is data connection (each of the three uses does not overlap with the other two). When web surfing 30 percent of the execution time is graphics, 50 percent is CPU use and 20 percent is data connection. The phone does web surfing about 30% of the time and video 70% (calls are ignored).

1. Currently web pages take (on the average) 8 seconds which we want to do in 5 seconds,
  - (a) If we only changed the CPU, how much faster would it need to be? (no other changes) (3 points)
  - (b) If we only changed the graphics, how much faster would it need to be? (3 points)
2. If we want the phone video performance to be 25% faster (speedup of 1.25), how can we do that? (4 points)