PRELIMINARY: Questions 1 and 2 refer to the following two pograms, which both do the same thing—the second program is just a modified ("loop-unrolled") version of the first.

```
r1: points to the beginning of vector A (constants)
r2: points to the beginning of vector B (input stream)
r3: points just past the end of vector A
r4: holds the running sum
r5-r8: used as scratch registers
```

PROGRAM FOO:

```
top:   lw     r5, r1, 0          // load into r5 from memory location (r1+0)
       lw     r6, r2, 0          // load into r6 from memory location (r2+0)
       mul    r5, r5, r6         // multiply r5 by r6, overwrite r5 with result
       add    r4, r4, r5         // add product (in r5) to a running sum in r4
       addi   r1, r1, 4          // increment A index by 4
       addi   r2, r2, 4          // increment B index by 4
       bne    r1, r3, top        // branch to "top" if A pointer is still valid
       exit                      // terminate program if A pointer is beyond end
```

PROGRAM BAR:

```
top:   lw     r5, r1, 0          // load into r4 from memory location (r1+0)
       lw     r6, r2, 0          // load into r5 from memory location (r2+0)
       lw     r7, r1, 4          // load into r6 from memory location (r1+4)
       lw     r8, r2, 4          // load into r7 from memory location (r2+4)
       mul    r5, r5, r6         // multiply r5 by r6, overwrite r5 with result
       mul    r7, r7, r8         // multiply r7 by r8, overwrite r7 with result
       add    r4, r4, r5         // add product (in r5) to a running sum in r4
       add    r4, r4, r7         // add product (in r7) to a running sum in r4
       addi   r1, r1, 8          // increment A index by 8
       addi   r2, r2, 8          // increment B index by 8
       bne    r1, r3, top        // branch to "top" if A pointer is still valid
       exit                      // terminate the program
```

1. Application Performance on a Pipeline (3 questions, 12 points)

A. (2pts) What do programs FOO and BAR do (i.e., what function/operation do they perform)?

B. (4pts) What is pipelining? What is the benefit of pipelining a microprocessor?

C. (6pts) Calculate the performance of both FOO and BAR on a pipelined processor, and be careful in your calculations. Assume that a **LW** (load-word) operation takes **4 cycles** but can be pipelined (a new LW issued every cycle). Assume that a **MUL** (multiply) operation takes **2 cycles** and can be pipelined just like the LW operation. All other operations (**ADD, ADDI, BNE**) execute in **1 cycle**. Assume perfect branch prediction. Give the running times for the two programs in the number of *cycles* required to execute. Assume the vector length is exactly one million 4-byte elements. You may ignore the cost of the "exit" instruction at the end of execution.

2. Caches (2 questions, 8 points)

A. (4pts) What is a cache? How and why does it work?

B. (4pts) What aspect/s of a cache's organization (choosing from cache size, cache block/line size, cache associativity, and replacement policy) would be most valuable in speeding up the **data** performance of programs FOO and BAR? Justify your answer (i.e., explain why).