The following is a list of possible questions for our quiz on September 5th. Some of the questions will not be asked in the quiz. All the questions that will appear in the quiz will appear exactly as shown below (however, numeric parameters may be changed). The quiz is closed textbook, closed notes and closed neighbors. Note that the questions, which did not appear in this quiz, still may appear in the exams. You will find a solution for these questions during lectures.

- **It is suggested that you bring your calculator** (you can use your calculator during the quiz on September 5th).

### Part I – the topics from August 24:

**#1:** Complete the following figure by specifying number systems we discussed in the classroom.

```
Arithmetic systems used in computers

Integers

Fractions
```

**#2:** Show “78\((10)\)” using unsigned integer format. How many bits are needed?

**#3:** What are the two major weaknesses in “unsigned integer”?

**#4:** Show the binary bit pattern for “-5” in 4-bit sign-magnitude integer. Show all your work.
#5: What does the following 16-bit two’s complement number mean in the decimal format? Show your calculation.

“1 0 1 1 1 0 0 0 0 1 0 0 1 0 1”

#6: What is “Overflow Flag” in a processor (how is it used) for handling operations on two’s complement integers?

#7: Transform the following decimal number to the two’s complement binary number (using the 16-bit format: your processor is a 16-bit architecture processor): -63_{(10)}. Show all your work.

#8: Which numbering system solves the two problems in the “sign magnitude integers”?

#9: How will “0_{(10)}” be represented using n-bits two’s complement number?

#10: How will “-1_{(10)}” be represented using n-bits two’s complement number?

#11: Show the binary bit pattern for “-5” in 4-bit two’s complement integer.

Part II – the topics from August 27:

#12: What are “registers” in processors?

#13: How are registers in processors used when a binary executable program is executed?

#14: What is “PC-SPIM”?

#15: Why do we need “jr $31” at the end of an assembly program?

#15: What are “system calls” in “MIPS Simulator”?

#16: “li $t0, (1024)” is an illegal instruction (if you try to assemble that instruction using PC-Spim simulator, that instruction will cause a syntax error). What’s wrong?

#17: What is the difference between “li $a0, 1024” and “la $a0, 1024” instructions? Assume that this computer system is a 32-bit system (i.e., all the registers are 32-bit registers and its ALU can deal with up to 32-bit inputs and outputs).

#18: “li $t1, $t0” is an illegal instruction (if you try to assemble that instruction using PC-SPIM simulator, that instruction will cause a syntax error). What’s wrong?