CS 061 – Computer Organization **Fall - 2013**

Quiz 1 – Thursday 10/3 solution

Each question is worth 1 point.

1. What is one difference between a High Level Language (HLL) like C++ and an Assembly Language (AL)?
   - **a. A single HLL statement might compile to many Machine Language instructions, while each AL instruction assembles to a single Machine Language instruction**
   - **b.** A single AL statement might compile to many Machine Language instructions, while each HLL instruction assembles to a single Machine Language instruction
   - **c.** AL programs are not as powerful as HLL programs
   - **d.** There is no major difference between them.

2. The LC-3 AL program you wrote in lab1 multiplied a number stored in a register by the constant 6. Which of the following best describes the algorithm used?
   - **a.** Use the LC-3 instruction MULTIPLY
   - **b.** Invoke the AL library subroutine MULTIPLY
   - **c. Add the value in the register to itself 6 times**
   - **d.** Look up a multiplication table stored in memory

3. If two microprocessors are separately designed & built to the specifications of the same ISA (Instruction Set Architecture), they will be *functionally* identical. They will also necessarily end up being *electronically* identical:
   - **a.** True
   - **b. False**

4. A label in assembly language code is:
   - **a.** An abbreviation for an instruction
   - **b.** Just a visual reminder for the programmer, ignored by the assembler.
   - **c.** A name given to a variable
   - **d. A name given to a memory location**
   - **e.** An adhesive sticker placed on the front page of the code

5. A "pseudo-op" in assembly language code (e.g. the LC-s's ".FILL") is:
   - **a. an instruction to the assembler (like a compiler preprocessor directive)**
   - **b.** an alias for a genuine assembly language instruction
   - **c.** a "place-holder" which has no effect on the code
   - **d.** a "redirect" placed in the code

6. Assembly language instructions can be categorized in three groups. These are:
   - **a. Operations, Data Movement and Control**
   - **b.** Labels, instructions, pseudo-ops
   - **c.** High Level, Assembly, and Machine
   - **d.** Signed magnitude, one's complement, and two's complement
   - **e.** Load, store, and arithmetic

7. The asssembly language instruction `ADD R0, R1, R2`
   - **a.** adds the contents of registers 0, 1 and 2, and stores the resutl in register 2
   - **b.** adds the contents of registers 0, 1 and 2, and stores the resutl in register 0
   - **c.** adds the contents of registers 0 and 1, and stores the resutl in register 2
   - **d. adds the contents of registers 1 and 2, and stores the resutl in register 0**
   - **e.** none of the above

8. The total "number of numbers" representable by a 9-bit binary word is:

a. 32 **c.** 128 **e.** **512**
b. 64 **d.** 256 **f.** 1024

9. Convert the 8-bit unsigned magnitude binary number `1000 1111` into decimal *("unsigned magnitude" is just the name of the encoding we studied on Tuesday)*

a. 15 **c.** 113 **e.** 255
b. 81 **d.** **143** **f.** 321

10. Convert the decimal number 53 into an 8-bit unsigned magnitude binary number:

a. `0000 1011` **c.** **`0011 0101`** **e.** `0011 0011`
b. `0101 0101` **d.** `0010 1101` **f.** `1000 1101`