Midterm Examination
CEG 320: Computer Org. & Assembly
REVIEW

NAME: _____________________________

I. Multiple Choice [70 pts]. Answer ALL of the questions in this section. Please select the best answer for each of the questions below.

Questions 1-5 use the contents of a LC-3 computer memory represented by the table, below.

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>x0</td>
<td>xFFFD</td>
</tr>
<tr>
<td>x1</td>
<td>xE000</td>
</tr>
<tr>
<td>x2</td>
<td>x0000</td>
</tr>
<tr>
<td>x3</td>
<td>xC024</td>
</tr>
<tr>
<td>x4</td>
<td>x0002</td>
</tr>
</tbody>
</table>

___ 1. The integer equivalent for the 2’s complement value stored at memory address x0 is: (a) -65,533, (b) -13, (c) -3, (d) 3, (e) 13, (f) 65,533, (g) None of the above.

___ 2. The character equivalent for the ASCII value stored at memory address x3 is: (a) 0, (b) 8, (c) 48, (d) H, (e) eot, (g) None of the above.

___ 3. The mixed fraction equivalent for the IEEE floating point value stored in x2 and x3 is: [Recall that in a little endian machine such as the LC-3, the least significant word is held the lowest address (x2) and the most significant word is held in the highest address (x3): (a) -5 1/8, (b) -2 9/16, (c) 2 9/16, (d) 5 1/8, (e) A very large positive number, (f) A very large negative number, (g) None of the above.

___ 4. If the contents of memory address x1 are loaded into the IR and executed as an instruction, what would be the new contents of R0? (a) x1, (b) x2, (c) x4, (d) x8, (e) No change, (d) None of the above

___ 5. A program can be translated into the ISA of a processor by means of: (a) a compiler, (b) an assembler, (c) an interpreter, (d) All of the above, (e) None of the above.

___ 6. The sequence of an Instruction cycle is: (a) FETCH -> DECODE -> EVALUATE ADDRESS -> FETCH OPERANDS -> EXECUTE -> STORE RESULT; (b) FETCH -> DECODE -> FETCH OPERANDS -> EVALUATE ADDRESS -> EXECUTE -> STORE RESULT; (c) FETCH -> DECODE -> FETCH OPERANDS -> EXECUTE -> EVALUATE ADDRESS -> STORE RESULT; (d) FETCH -> FETCH OPERANDS -> DECODE -> EVALUATE ADDRESS -> EXECUTE -> STORE RESULT; (e) None of the above
7. Which of the following instructions can reference a memory location that is #1000 locations from the instruction? (a) ADD, (b) LD, (c) STR, (d) LEA, (e) All of the above, (f) None of the above

8. The addressability of LC-3 memory is: (a) 2^16 locations, (b) 16 bits, (c) 2^9 locations, (d) 2^16 bits, (e) None of the above

9. After the execution of which of the instructions below will the value in the Program Counter (PC) be 0x306e? (a) i, (b) ii, (c) iii, (d) i and ii, (e) i and iii, (f) ii and iii, (g) i, ii, and iii, (h) None of the above
   
   i. 0000111001011100 at location 0x3011
   ii. 0000000001101110 at location 0x306d
   iii. 1010011001101110 at location 0x306d

10. In a 16-bit address space A[15:0], if A[15:12] = 1111 were allocated to I/O device registers, the number of addresses assigned to memory mapped I/O is: (a) 0, (b) 1k, (c) 2k, (d) 4k, (e) 8k

11. The instruction TRAP x23 at location x4232, causes R7 to be loaded with: (a) x0023, (b) x4232, (c) x4233, (d) contents of x0023, (e) contents of x4232

12. If the condition codes have values N=0, Z=0, P=1 at the beginning of the execution of the following sequence of instructions, what will their values be at the end of the execution of the following sequence of LC-3 instructions? Assume all registers hold values of x0000 when entering this segment of code.
   
   x3050 0000 0010 0000 0001
   x3051 0101 0000 0010 0000
   x3052 0001 0000 0010 0001

   (a) N=0, Z=0, P=1; (b) N=0, Z=1, P=0; (c) N=1, Z=0, P=0; (d) N=0, Z=0, P=0; (e) None of the above

13. In a C/C++ program, we encounter the integer value x (previously declared with the statement “int x;”) in the conditional statement: “if (abs(2 * x) >= abs(x)) {foo();} else {bar();}”. Which of the following statements is most true: (a) foo() will always be called; (b) bar() will always be called; (c) For most values of x, foo() will be called, but bar()_might_ be called for one or more values of x; (d) For most values of x, bar() will be called, but foo() _might be called for one or more values of x; (e) foo() and bar() are equally likely to be called.

14. In a C/C++ program, we encounter the floating-point value f (previously declared with the statement “float x;”) in the conditional statement: “if (-1.5 * (-1.5 * f)) >= ((-1.5*-1.5) * f) {foo();} else {bar();}”. Which of the following statements is most true: (a) foo() will always be called; (b) bar() will always be called; (c) For most values of f, foo() will be called, but bar()_might_ be called for one or more values of f; (d) For most values of f, bar() will be called, but foo() _might be called for one or more values of f; (e) foo() and bar() are equally likely to be called.
II. Practical Programming

1. [30 pts] Consider the following segment of “buggy” assembly code. The intent of this program is to add together two number (2 and 3) and then to display the result to the display. The program has several errors.

```
.ORIG x3000
AND R0,R0, #0
AND R2,R2, #0
ADD R2,R2, #2
JSR SUB
LD  R7, ASCII
ADD R2,R2, R7
ADD R0,R2,#0
TRAP x21 ; TRAP OUT; DISPLAY <- R0
TRAP x25 ; TRAP HALT
SUB ADD R2,R2,#3
ADD R7,R7,#1
RET
ASCII .FILL x0030
.END
```

(a) [10] Provide the symbol table for this segment of code.

(b) [10] What does this program _actually_ do, as written? Be specific. What will be the _actual_ output behavior of this code segment when run? How does that result come about?

(c) [10] Rewrite the assembly code to “fix” the bugs.