Final (Fall 2002), CSCI 150, Fall 2003

Name: _____

1. **a.** Give a sum-of-products Boolean expression equivalent to the following truth table.

x	y	z	output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

b. Simplify this expression.

c. Draw a circuit that computes the simplified expression.

2. Define the fetch-execute cycle as it relates to a computer processing a program. Your definition should include an explanation of the primary purpose of the fetch phase and the execute phase.

3. Suppose we were to execute the following method at right, passing $\langle 5, 2, 7, 0, 6, 1 \rangle$ for the array parameter `intArray`.

a. Show the sequence of values taken by the variables `k`, `sk`, and `i`.

```
k
sk
i
```

b. What value does the method return?

```
public static int myst(int[] intArray) {
    int k;
    int sk;
    if(intArray[0] > intArray[1]) {
        k = intArray[0];
        sk = intArray[1];
    } else {
        k = intArray[1];
        sk = intArray[0];
    }
    int i = 2;
    while(i < intArray.length) {
        if(intArray[i] > k) {
            sk = k;
            k = intArray[i];
        } else if(intArray[i] > sk) {
            sk = intArray[i];
        }
        i++;
    }
    return sk;
}
```

4. The Java program at right inputs an integer, reads in that many integers. and then displays how many of them exceed 5. Translate this into an equivalent assembler language program for HYMN.

```
import csbsju.cs150.*;
public class Test {
    public static void main(String [] args) {
        IOWindow win = new IOWindow();
        int left = win.readInt();
        int count = 0;
        while(left > 0) {
            int value = win.readInt();
            if(value > 5) {
                count = count + 1;
            }
            left = left - 1;
        }
        win.println(count);
    }
}
```

5. *Covers material not covered this semester. The question involved tracing a program in last year's assembly language. It doesn't translate directly into HYMN's assembly language.*

6. The game of Nim proceeds by players taking turns selecting a pile and removing stones from that pile. The player removing the last stone wins.

Draw a complete game tree for the game of Nim beginning with two piles, both containing two stones. To draw a node, list the number of stones in each pile; for example, the top node will be "2,2."

Do not include the minimax values assigned to each node in your tree.

7. Describe the Turing Test and its purpose.

8. **a.** Design a finite state automaton that will recognize the language of all strings containing only *a*'s and *b*'s where there are at least 3 *b*'s.

b. Write a regular expression for the language consisting of all strings which either have at least one *a* or contain the string *bcd*. As with *egrep*, you may use a period ("`.`") to represent any single character.

9. Design a Turing machine that transforms a string containing only *a*'s, *b*'s, and *c*'s by replacing each letter preceding an *a* to a *b*. (Do not worry about the case when the string begins with an *a*.) Thus, *bccb* would remain unchanged while *caccaa* would be changed to *bacbba*. The Turing machine should always eventually enter an accepting state to terminate.

10. Perform the following conversions.

a. $10101_{(2)}$ to decimal.

b. $38_{(10)}$ to binary.

c. $1101011100_{(2)}$ to hexadecimal.

d. $\text{A3D}_{(16)}$ to binary.

11. *Covers material not covered this semester.*

12. *Covers material not covered this semester.*

code	op	behavior
000	HALT	nothing further happens (computer halts)
001	JUMP	PC $\leftarrow$ $data$
010	JZER	**if** AC $= 0$ **then** PC $\leftarrow$ $data$ **else** PC $\leftarrow$ PC $+ 1$
011	JPOS	**if** AC > 0 **then** PC $\leftarrow$ $data$ **else** PC $\leftarrow$ PC $+ 1$
100	LOAD	AC $\leftarrow$ $\mathbf{M}[data]$; PC $\leftarrow$ PC $+ 1$
101	STORE	$\mathbf{M}[data]$ $\leftarrow$ AC; PC $\leftarrow$ PC $+ 1$
110	ADD	AC $\leftarrow$ AC $+ \mathbf{M}[data]$; PC $\leftarrow$ PC $+ 1$
111	SUB	AC $\leftarrow$ AC $- \mathbf{M}[data]$; PC $\leftarrow$ PC $+ 1$

Solutions, Final (Fall 2002), CSCI 150, Fall 2003

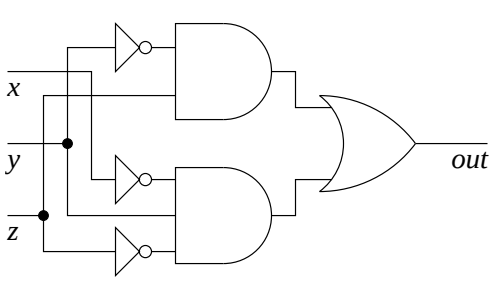
mean	89.350 (1787.000/20)
stddev	21.987
median	92.000
midrange	70.500-108.500

#	avg	#	avg
1	9.10 / 10	8a	3.75 / 5
2	6.45 / 10	8b	2.90 / 5
3	7.45 / 10	9	6.60 / 10
4	5.50 / 10	10	9.65 / 10
5	7.25 / 10	11	6.55 / 10
6	7.15 / 10	12	8.75 / 10
7	8.25 / 10		

1. **a.** $\bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z$

b. $\bar{y}z + \bar{x}y\bar{z}$

c.



2. The fetch-execute cycle is the process by which a classical computer executes instructions. In the fetch phase, the computer determines the next instruction to be completed by fetching the instruction from memory based on the address found in the program counter. In the execute phase, the computer executes the instruction just fetched.

3. **a.**

```
k    5 7
sk   2 5 6
i    2 3 4 5 6
```

b. 6

4.

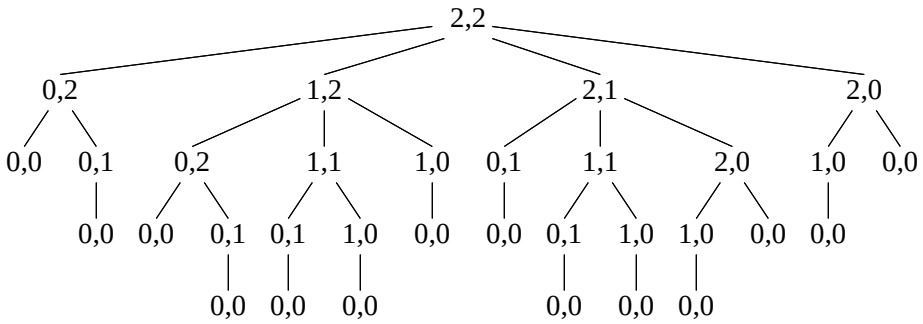
```
      READ        # read left from user
      STOR left
loop: LOAD left   # if no more are left, we are done
      BRZR done
      READ        # read user's number and subtract 5
      SUB five
      BRNG yes    # if it's negative, user's number is < 5
      JUMP no     # otherwise, jump past if
yes:  LOAD count  # increment count
      ADD one
      STOR count
no:   LOAD left   # decrement left
      SUB one
      STOR left
      JUMP loop   # go again
```

```
done: LOAD count  # display count
      WRITE
      HALT

count: 0
left:  0
one:   1
five:  5
```

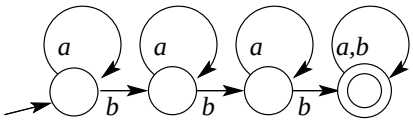
5. *Covers material not covered this semester.*

6.



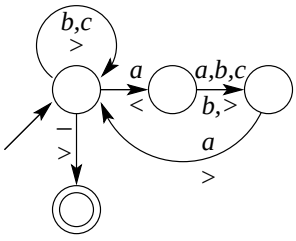
7. In the Turing Test, a human tester sends questions and receives answers from a human and a computer hiding in the other room, and the tester tries to determine which is the human and which is the computer. If the tester cannot tell, then the computer is deemed intelligent.

8. **a.**



b. `.*(a|bcd).*`

9.



10. **a.** $21_{(10)}$

b. $100110_{(2)}$

c. $35C_{(16)}$

d. $101000111101_{(2)}$

11. *Covers material not covered this semester.*

12. *Covers material not covered this semester.*