15-110: PRINCIPLES OF COMPUTING – EXAM 2 SAMPLE – FALL 2014

Name ___________SOLUTIONS_________________ Section _________

Directions: Answer each question neatly in the space provided.

Please read each question carefully. You have 50 minutes for

this exam. No electronic devices allowed. Good luck!

1. The following question deals with recursion and recursive algorithms.

1.a. (6 pts) The function *f* is defined for non-negative integers *a* and *b* recursively as follows:

$$f(a,b) = \begin{cases} 0 & if\ b = 0 \\ f(a, b-1) + a & if\ b > 0 \end{cases}$$

Compute f(4, 3) by drawing a recursion tree showing all of the computation required and then use your tree to compute the answer.

Recursion Tree: *f*(4, 3) = _______12_______

```
f(4, 3) = f(4, 2) + 4  = 8 + 4 = 12
   |
f(4, 2) = f(4, 1) + 4 = 4 + 4 = 8
   |
f(4, 1) = f(4, 0) + 4 = 0 + 4 = 4
   |
f(4, 0) = 0
```

State the mathematical function that *f* is performing (i.e. a+b, HINT: That's not the answer).
_______f(a, b) = a * b_______

1.b. (4 pts) Suppose you want to look up a phone number in a telephone book which has 1000 pages. For this problem, consider only finding the correct page in the book by comparing the name you are looking for to the first name on any given page. If you use binary search, how many name comparisons must you perform (approximately) in the worst case?

______$\log_2 1000 = 10$______

An alternative is to use linear search. How many name comparisons must you perform, in the worst case, using linear search?

__________1000_________

1.c. (6 pts) Complete the following Python function recursively so that it finds the largest integer in list. You may assume list has a length of at least 1. Do not use a loop in your answer.

```
def largest(list):
     if len(list) == 1:
          return list[0]
     else:
          return max(list[0], largest(list[1:]))
```

2. This question concerns generating random numbers.

Recall that the Python function `randint(0, n)` returns a random integer between 0 and n, inclusive. Using the `randint` function, show how to compute the following using one Python expression:

2a. [2 pts] A random integer between 10 and 20, including 10 and 20.

_____randint(10, 20)_____

2b. [2 pts] A random **even** integer between 10 and 20, including 10 and 20.

_____randint(5, 10) * 2____

2c. [2 pts] A random **odd** integer between 5 and 15, including 5 and 15.

__randint(2, 7) * 2 + 1____

2d. [3 pts] A random string from the array `colors` below. You are **not** allowed to use any function other than `randint` from the module `random`.

____colors[randint(0,3)]___

```
colors = ["red", "green", "blue", "orange"]
```

2b. [5 pts] Write a function called `fair_coin()` that simulates flipping a fair coin. Your function should return a string. **Hint:** In a fair coin, the outcomes "head" and "tail" are equally likely.

```
def fair_coin():
    return [“head”, “tail”][randint(0, 1)]
```

```
def fair_coin():
    if randint(0, 1) == 0:
        return “head”
    else:
        return “tail”
```

2c. [6 pts] Write a function called `biased_coin()` that simulates flipping a biased coin that returns "head" 80% of the time, and "tail" the rest of the time.

```
def biased_coin():
    if randint(0, 99) < 80:
        return “head”
    else:
        return “tail”
```

3. This problem focuses on the representation of data in a computer.
The following tables may be helpful in this question:

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

Bin	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

3.a. (2 pts) Compute the decimal value of the byte 10101010
if it is interpreted as an unsigned integer.

$2^7 + 2^5 + 2^3 + 2^1 = 170$

_________170_________

3.b. (2 pts) Compute the decimal value of the byte 10101010
if it is interpreted as a signed 2's complement integer.

10101010 => 01010101 + 1 => 01010110 = $2^6 + 2^4 + 2^2 + 2^1 = -86$ _________-86___________

3.c. (2 pts) The ASCII character 'A' is represented in binary using 7 bits as 1000001. The character is to be sent via satellite using *odd parity*. What eighth bit is sent along with this byte: 1 or 0?

__________1__________

3.d. (2 pts) Suppose that the eighth (parity) bit is corrupted during transmission
of the eight bits from part (d) and is "flipped" (either from 0 to 1 or 1 to 0).
Which of the following is true? Select the appropriate letter and write it here:

(A) The receiver cannot detect the error.
(B) The receiver can detect the error but cannot determine which bit is wrong.
(C) The receiver can detect the error and can correct the bit that is wrong.

__________B__________

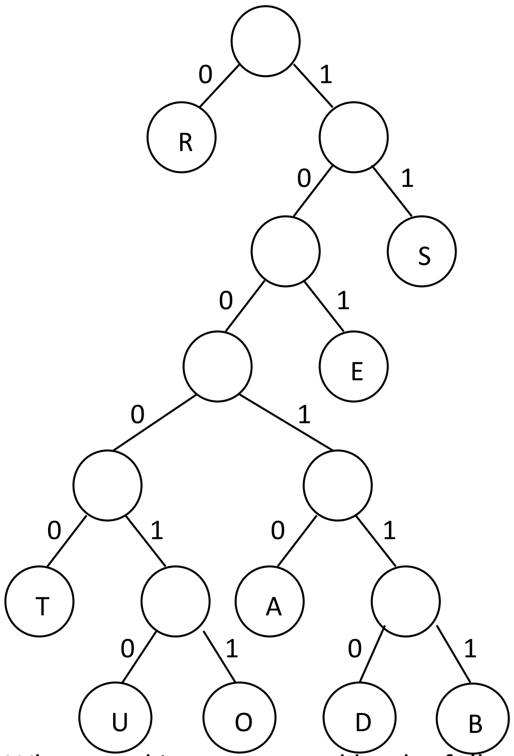
3.e. (2 pt) Entropy is defined as the average amount of information contained in a data stream. What is the entropy for a data stream of 128 possible 7-bit ASCII codes, each of which is equally as likely?

__________7____________

3.f. (1 pt) Suppose we were to add a parity bit to the data stream described above such that we have 256 possible 8-bit ASCII codes. (Remember what the parity bit is supposed to do!). If the error rate of the data is relatively low, will the entropy be relatively low or relatively high?

Relatively low

3.h. (4 pts) Based on the following Huffman tree:



What word is represented by the following binary string based on the Huffman tree:

100110|101|10010|0|101|11|10000 ________DEAREST___________

Suppose we want to encode words made using the nine letters from the tree above using a *fixed-width encoding* with the fewest bits possible for each letter.
How many bits are required to encode each letter? ____________4______________

2^3 = 8 unique letters, which is not enough => 2^4 = 16 unique letters

4. This question deals with binary trees and graphs.

4.a. (6 pts) Draw the binary search tree that results by inserting the following integers into the tree in the order shown. Show the tree at each insertion step. That is, you will draw 7 non-empty trees.

4 2 1 3 6 5 7

```
4   4       4       4        4          4           4
   /       /       /        / \        /   \       /   \
  2       2       2        2   6      2     6     2     6
         /       / \      / \        / \   /     / \   / \
        1       1   3    1   3      1   3 5     1   3 5   7
```

4.b. (4 pts) Write a Python list that represents the binary tree from part 4a. Note that this can be done in one of several ways. You can use the representation from the assignments or something else. Write clearly how you intend us to interpret your representation. How do you represent the root, the left and right subtrees?

[value, [left subtree], [right subtree]] => [4, [2, [1, [], []], [3, [], []]], [6, [5, [], []], [7, [], []]]] or

[[left subtree], value, [right subtree]] => [[[[], 1, []], 2, [[], 3, []]], 4, [[[], 5, []], 6, [[], 7, []]]]

4.c. (2 pts) How many comparisons are needed to find the item 7 in the binary search tree from part 4a, starting from the root and using the binary search algorithm? List all the key values that 7 is compared to until it is found.

4, 6, 7

4.d. (3 pts) If you have a binary search tree with *n* nodes, what is the minimum number of levels in that tree?

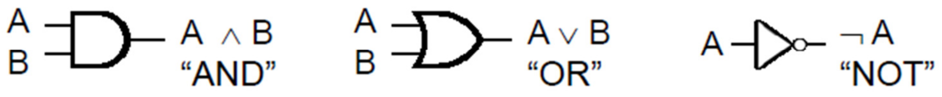
log(n)

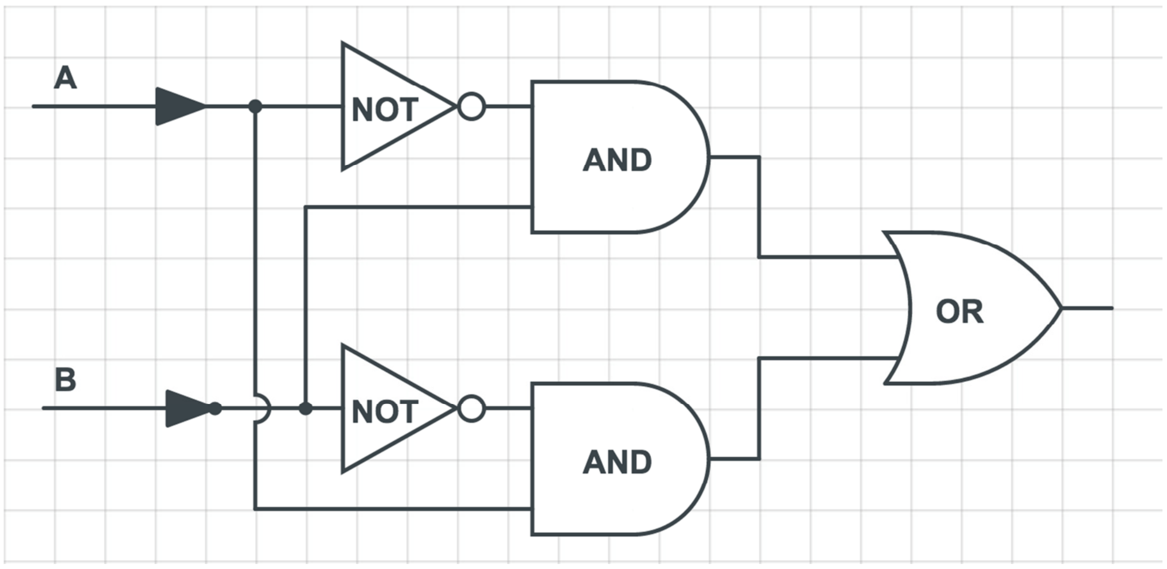
5. The following question involves Boolean logic and abstraction.

5.a. (6 pts) Let S = $(A \wedge \neg B) \vee (\neg A \wedge B)$, where A, B and C are Boolean variables. Fill in the truth table below to compute S.

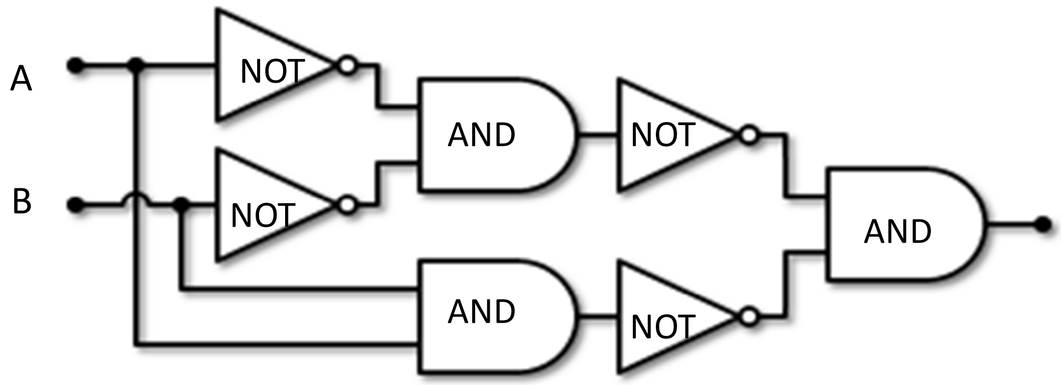
A	B	$A \wedge \neg B$	$\neg A \wedge B$	S
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

5.b. (4 pts) The Boolean value S from part 5a can be computed by an electronic circuit. Draw this circuit at the gate level of abstraction. Below is a reminder for the diagrams for AND, OR and NOT gates.





5.c. (6 pts) Consider the following circuit. Write the Boolean expression that represents the circuit below.



$\neg(\neg A \wedge \neg B) \wedge \neg(A \wedge B)$

5.d. (4 pts) Using the De Morgan's Law, rewrite the expression `not(x < 3 or x >= 4)`.

(x >= 3) and (x < 4)