15110 PRINCIPLES OF COMPUTING – EXAM 2- Summer 2015

Q. 1	[15]	_____
Q. 2	[20]	_____
Q. 3	[20]	_____
Q. 4	[15]	_____
Q. 5	[21]	_____
Q. 6	[9]	_____
TOTAL	**[100]**	_____

Name: ANSWER KEY **Section:** _____

Andrew Id: _____

Directions: Answer each question neatly in the space provided.
Please read each question carefully. You have **80 minutes** for this exam.
No electronic devices allowed. Good luck!

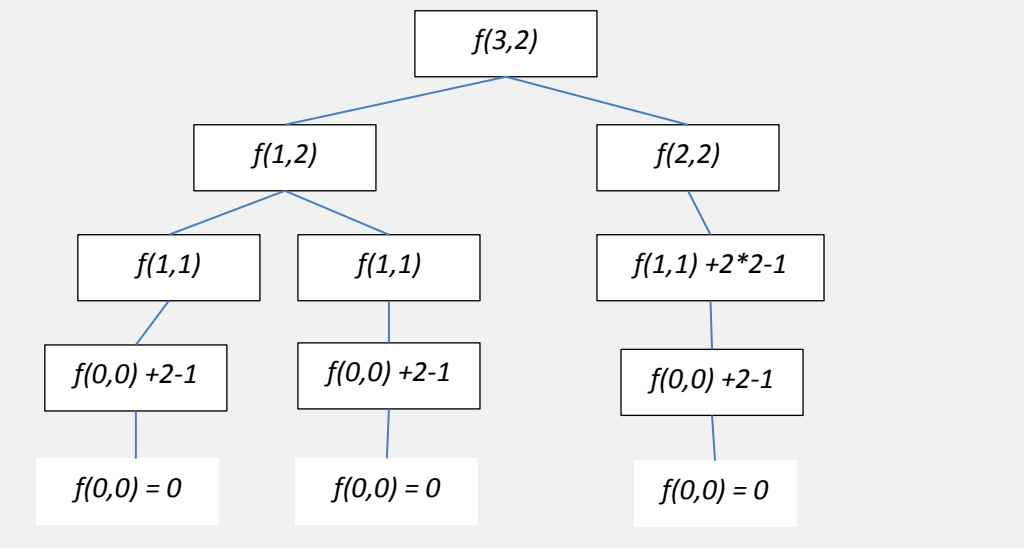
1. The following question deals with recursion and recursive algorithms [15 points]

(a) [7 pt] The function *f* is defined for **non-negative** integers *a* and *b* recursively as follows:

$$f(a,b) = \begin{cases} 0 & if\ a\ = 0\ or\ b = 0 \\ f(a-1,b-1) + 2a - 1 & if\ a = b \\ f(a-b,b) + f(b,b) & if\ a > b \\ f(a,a) + f(b-a,a) & if\ a < b \end{cases}$$

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*(3, 2) = 6



State the common name for *f* or write a very compact non-recursive definition of *f*:

multiply → f(a, b) = a * b

(b) [5 pts] Complete the recursive function to return a list that consists of the negative numbers of a given list (e.g. negatives([6, 3, -4, 9, -11, -2, 5]) will return [-4, -11, -2])

```
def negatives(lst):
    if len(lst) = 0 :
        return  []
    else:
        if lst[0] < 0 :
            return [lst[0]] + negatives(lst[1:])
        else:
            return  negatives(lst[1:])
```

(c) [3 pts] Write the formula that `fMystery` function calculates.
(Hint: you can test it with a few numbers)

```
def fMystery(a):
    if a == 0:
        return 0
    else:
        return fMystery(a-1) + 2*a-1
```

It calculates the square of a given number:

fMystery(a) calculates a^2

2. This problem focuses on representation of data in a computer. [20 points]

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

(a) [2 pts] Compute the decimal value of the byte **10101110** if it is interpreted as an unsigned integer.

$2^7+2^5+2^3+2^2+2^1 = 128+32+8+4+2 = 174$

(b) [2 pts] Compute the decimal value of the byte **10101110** if it is interpreted as a signed 2's complement integer.

1 0101110 → (flip and add 1)

0 1010001 + 1→ 1010010 = $2^6 + 2^4 + 2^1$ = 64+16+2 = -82

(c) [2 pts] Express the byte **10101101** in hexadecimal.

10101101 → 1010 1101 → AD in Hexadecimal (= 0xAD)

(d) [2 pts] The ASCII character 'Q' is represented in binary using 7 bits as 1010001. The character is to be sent via satellite using even parity. What eighth bit is sent along with this byte: 1 or 0?

1 *(because there are 3 ones and to check with even parity 1 is used as parity check to make it even)*

(e) [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:

B *(11010001 will become 01010001)*

(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.

(f) [2 pts] In an HTML file for a webpage, the designer used the following font tag that changes the color of the font based on the 6 digit hexadecimal value 876543.

This is a colorful sentence.

The 6 digit hexadecimal value specifies the amount of red, green and blue for the font's color, respectively (in the format RRGGBB). Express the amount of green in the font as an integer between 0 and 255, inclusive. Show your work.

65 Hexadecimal = 6 * 16 + 5 = 96 + 5 = 101
(65 Hexadecimal is 01100101 in binary)

(g) [2 pts] Calculate the sum of the following three **binary** numbers (which represents unsigned integers): `00111, 00101, 01100`.

```
  00111
  00101
+ 01100
  11000
```

(h) [1 pts] Add the following three **hexadecimal** numbers together: 0x13, 0x98
(Note that "0x" is used before hexadecimal numbers as a convention.)

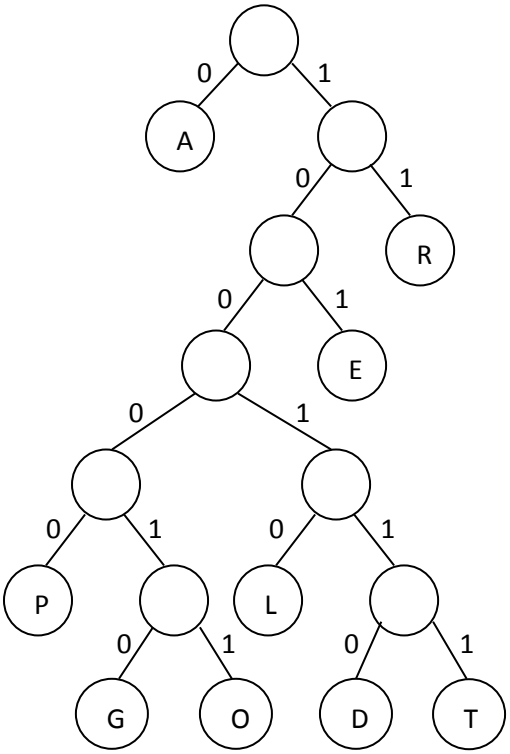
```
  0x13
+ 0x98

  0xAB
```

(g) [1 pts] Is MP3 (MPEG3) compression of sound files a *lossless* or *lossy* compression algorithm? **Lossy**

(h) [1 pts] Is JPEG encoding of graphics files a *lossless* or *lossy* compression algorithm? **Lossy**

(i) [3 pts] Based on the following Huffman tree:



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

100010111010100111 **GREAT**

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 bits (9 letters)**

How many bits are required to (re)encode the English word you decoded above, using a *fixed-width encoding*? **5 x 4 = 20 bits**

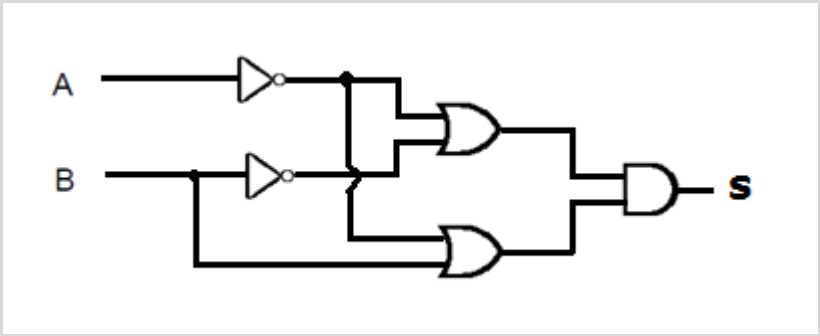
3. The following question involves Boolean logic and digital circuitry. [20 pts]

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

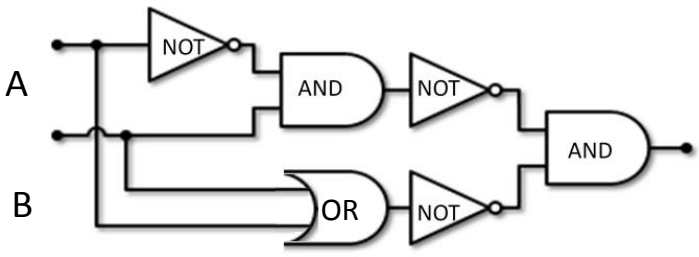
A	B	$\neg A \vee \neg B$	$\neg A \vee B$	S
0	0	1	1	1
0	1	1	1	1
1	0	1	0	0
1	1	0	1	0

(b) [4 pts] The Boolean value S from part 3a 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.

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



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



Not (Not A and B) and (Not (A or B)) **$(\neg A \vee \neg B) \wedge (\neg A \vee B)$**

(Not A NAND B) and (A NOR B) ← *This is also acceptable*

(d) [4 pts] Using the De Morgan's Law, rewrite the equivalent expression

```
not(5 >= y and x != 4)
```

5 < y or x == 4

4. This problem focuses on arrays, lists, stacks, and hash tables. [15 pts]

(a) [2 pts] If you have an array of 7500 elements in computer memory, and you want to insert an element at the beginning of the array, how many elements of the array need to be moved?

7500

(b) [2 pts] If you have a linked list of 7500 elements in computer memory, and you want to insert an element at the beginning of the linked list, how many nodes of the list need to be moved?

0 (No change of nodes, just 2 pointers will be changed)

(c) [3 pts] A hash table is used to store keys so we can search for them later. Suppose that keys are integers. Consider the following code of a hash function:

```
def hash(key):
    return key * 7 % 5
```

Consider the following set of keys:
3, 6, 11, 12.

What bucket does **each key** fall into?	Are there any **collisions**? Explain briefly.	How many buckets **in total** are there in this hash function?
3 → bucket 1 **6 → bucket 2** **11 → bucket 2** **12 → bucket 4**	**There is a collision. 6 and 11 both goes to bucket 2.**	**5 buckets in total (0, 1, 2, 3, 4) (because of %5)**

(d) [5 pts] Show how to compute the following RPN expression using a stack, showing the contents of the stack each time something is pushed or popped. The first three stacks are shown for you. Use as many columns as you need.

```
8 1 9 + 3 - 13 9 - * 20 - %
```

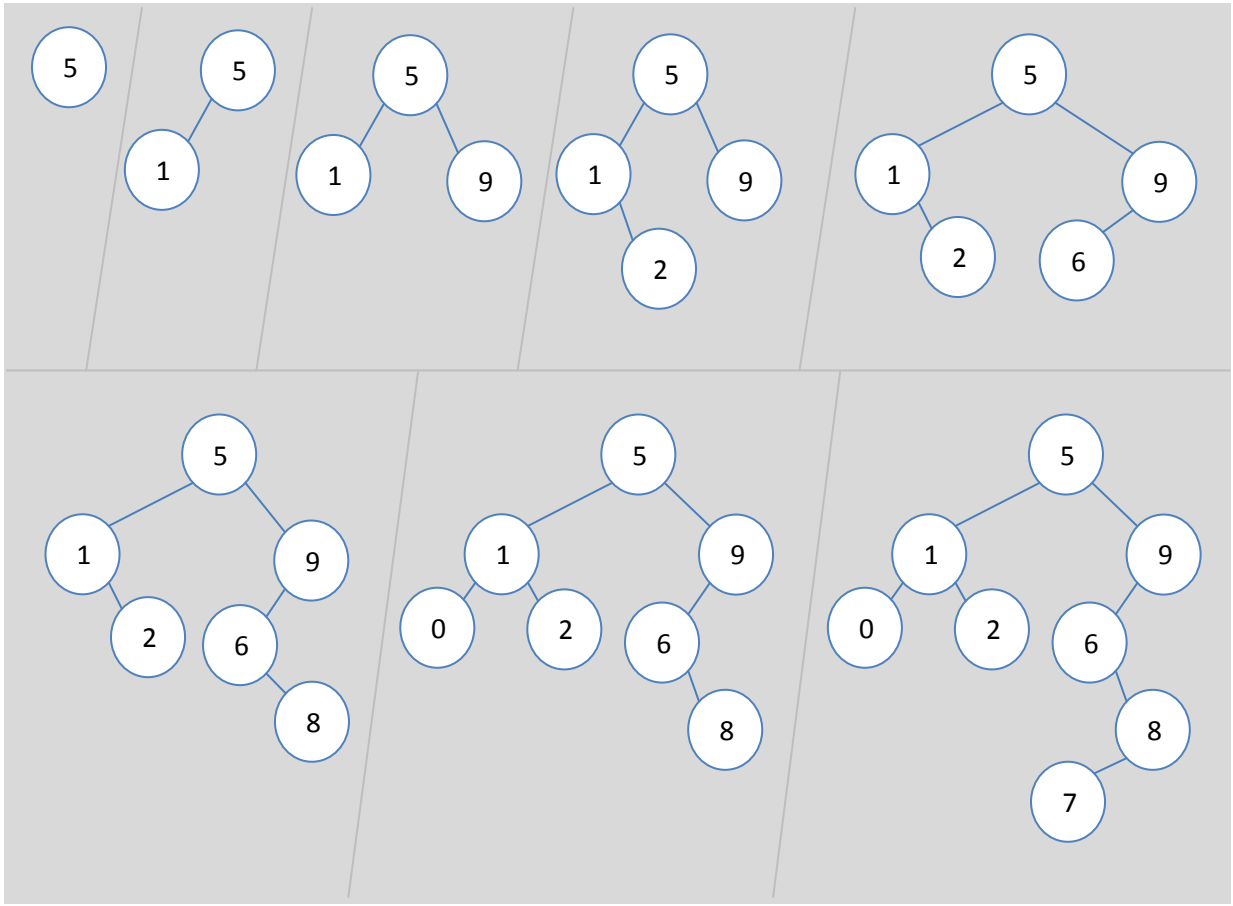
8	1 8	9 1 8	10 8	3 10 8	7 8	13 7 8	9 13 7 8	4 7 8	28 8	20 28 8	8 8	0		

(e) [3 pts] Suppose that algorithms A and B are two algorithms that can be used to search for a given word in an unsorted collection of 15000 words. Algorithm A operates on a hash table with 500 buckets where the words have been evenly distributed by a hash function, and Algorithm B operates on an array of size 15000 where each array position holds a single word. Which algorithm would be faster in searching for a word that does not exist in the list? Why?

Algorithm **A** would be faster because **each of 500 buckets will have 30 words. After using the hash function only 30 words will be checked. Algorithm B needs to check 15000 items**.

5. This question deals with binary trees and graphs. **[21 pts]**

(a) [4 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**. You will draw 8 non-empty trees. `5 1 9 2 6 8 0 7`



(b) [4 pts] Write a Python list that represents the binary tree from part 5a. 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 subtrees, and the right subtrees?

```
[    5,
     [1, [0, [], []], [2, [], []] ],
     [9, [6, [], [8, [7, [], []], []]], [] ]
]
```

Root

Left subtree

Right subtree

As a general rule [root, [left subtree], [right subtree]]

(c) [2 pts] How many comparisons are needed to find the item 8 in the binary search tree from part 5a, starting from the root and using the binary search algorithm? List all the key values that 8 is compared to until it is found.

4 comparisons (5, 9, 6, 8)

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

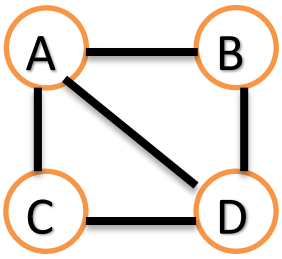
Floor($\log_2$(n)) + 1 ($\log_2$n + 1 or log(n)+1 is acceptable)

(e) [2 pts] Re-order the keys from part 5a (5 1 9 2 6 8 0 7) so that the binary search tree will have **at most 1 child per node**.

0 1 2 5 6 7 8 9

OR 9 8 7 6 5 2 1 0

(f) [4 pts] Write down the 4 by 4 adjacency matrix for the undirected graph below. Use the Boolean values True and False to indicate whether an edge exists between two nodes. For example, the matrix entry for row A and column C must be True.



	A	B	C	D
A	False	True	True	True
B	True	False	False	True
C	True	False	False	True
D	True	True	True	False

(g) [2 pts] Suppose that the graph from part 5f represents a social network where the nodes stand for people and edges stand for friendships. According to this graph, who has the fewest friends (A, B, C, or D)?

B and C have the fewest friends

6. This question deals with random number generators. [9 pts]

(a) [5 pts] Recall that the Python `randint(0,n)` function returns a random integer between 0 and `n`, inclusive. Using the `randint` function with 0 as its first argument, show how to compute the following:

A random integer from the set {0, 1, 2, 3} without storing these values in a list first

`randint(0, 3)`

A random integer form the set {11, 12, 13} without storing these values in a list first.

`randint(0, 2) + 11`

A random integer from the set {-4, -2, 0, 2, 4} without storing these values in a list first.

`randint(0, 4) * 2 – 4` **OR** **`(randint(0, 4) – 2) * 2`**

Write a function to return a random integer which should be square of a number and less than 150 such as `{1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144}`.

`def randSquare():` **`  return (randint(0, 11) + 1) ** 2`**	
`def randSquare():` **`  return randint(1, 12) ** 2`**	**`def randSquare2():`** **`  return randrange(1, 13) ** 2`**

(b) [4 pts] Consider the following Python functions:

`def rng1(n):`
`  return (n * 2) % 6`
What is the PRNG period of rng1?
 	
This is not a *Linear congruential generator*. To be a LCG it should be $x_{i+1} = (a\ x_i + c)\ \%\ m$. Starting with 0 it always returns 0. (1 or 0 is acceptable as your answers)
 	
Write the sequence of numbers starting at 0 until a repeat.
0 0…
	`def rng2(n):`
	`  return (3*n + 1) % 7`
	
	What is the PRNG period of rng2?
	
	6
	
	Write the sequence of numbers starting at 0 until a repeat.
	0 1 4 6 5 2 0 1 4…