

15110 PRINCIPLES OF COMPUTING – FINAL EXAM – FALL 2012

Name _____ Section _____ Andrew ID _____

Directions: Answer each question neatly in the space provided. Please read each question carefully. You have 180 minutes for this exam. No electronic devices or notes are allowed. Good luck!

	POINTS	GRADER
1	_____	_____
2	_____	_____
3	_____	_____
4	_____	_____
5	_____	_____
6	_____	_____
7	_____	_____
8	_____	_____
9	_____	_____
10	_____	_____
BONUS	_____	
TOTAL	_____	

1. (10 pts)

1a. Compute the value of each of the following Ruby expressions.

`10 * 8 + 5 - 6 / 3`

`(73 / 10) + (73 % 10)`

1b. Suppose you have a Ruby function `student_count(class, section)` that returns the number of students in a given class `class` and section `section`. Assume that the parameters `class` and `section` have the type integer. Write an assignment expression that uses the function `student_count` and gives the sum of the number of students in the class 15110 in sections 1 and 2.

`total_student_count = _____`

1c. The following Ruby function `print_elems_1(mylist)`, which includes a while loop, can be written using a for loop instead. Fill in the blanks in the function `print_elems_2(mylist)` such that it is equivalent to `print_elems_1(mylist)`, which means that for a given argument `mylist` it would print the same numbers as `print_elems_1(mylist)`.

```
def print_elems_1(mylist)
  i = 0
  len = mylist.length
  while i < len do
    print mylist[i], "\n"
    i = i + 1
  end
end
```

```
def print_elems_2(mylist)
  for _____ in _____ do
    _____
  end
end
```

1d. Let `colors = ["Red", "Blue", "Green", "Purple", "White", "Black"]`

What is the output in irb for each of the following Ruby code fragments?

```
colors.each { |item| print item }
```

```
colors.each { |item| print item if item > "Blue" }
```

1e. Write a Ruby function `create_table()` to create and return a two-dimensional array named `table` with 100 rows and 100 columns such that all the elements in the `table` are initialized to 0. Recall that you can create a one-dimensional array of size `n` using the method `Array.new(n)`.

2. (10 pts)

2a. Recall the algorithm merge sort for sorting elements in a list, which works by first dividing the input array into several chunks, and then combining them into bigger and bigger groups until the final merged group includes the full array. The function `merge_groups(a, gs)` below is a function that can be used to merge all adjacent groups of size `gs` to form groups of size `2*gs`.

```
def merge_groups(a, gs)
  i = 0                                # start of first group
  while i < a.length
    j = i + (2 * gs) - 1                # end of second group
    a[i..j] = merge_pair(a, i, gs)
    i = i + (2*gs)                      # start of next group
  end
```

Assume that the function `merge_pair(a, i, gs)` does the actual merge operation by merging the two groups at `a[i]` and `a[i+gs]`, and returning a new array with all the items in the merged group sorted in ascending order. Note that `a` stands for the array to be sorted, and `gs` for the group size. That is, an assignment statement such as `a[i..j] = merge_pair(a, i, gs)` tells Ruby to replace the current items in locations `i` through `j` of the array `a` with the array returned by `merge_pair(a, i, gs)`.

(i) If `a = [3, 8, 6, 7, 1, 9, 2, 10]`, what are the contents of `a[0..3]` right after the assignment statement `a[0..3] = merge_pair(a, 0, 2)` is executed?

`a[0..3] = _____`

(ii) If `a = [3, 6, 7, 8, 1, 9, 2, 10]`, what are the contents of `a[4..7]` right after the assignment statement `a[4..7] = merge_pair(a, 4, 2)` is executed?

`a[4..7] = _____`

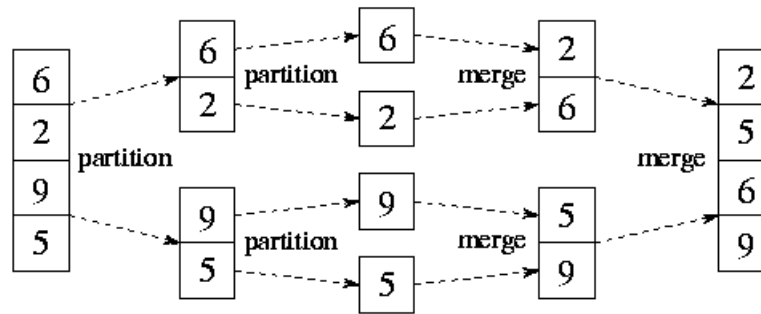
2b. Below is a Ruby function for a non-recursive implementation of merge sort that calls the `merge_groups(a, gs)` function from part (a). Complete the missing parts in the code.

```
def msort(array)
  a = array.clone    #creates an identical copy of array and stores it in a
  size = 1
  while size < a.length
    merge_groups(____, ____ )
    size = size * 2
  end
  return a
end
```

2c. Given the function `msort(array)` in part 2b, how many times would the function `merge_groups(a, gs)` be called to sort an array of 8 elements? _____ times.

2d. What is the worst-case order of complexity of this merge sort for n elements using the big O notation? You can assume that the complexity of `merge_groups(a, gs)` is linear in the number of elements in `a`. $O(\text{_____})$.

2e. Merge sort is an algorithm that lends itself to parallel execution of certain steps, where each step is executed on a different processor. Which steps of the algorithm can be executed in parallel (simultaneously)? Give your answer using the following figure as a reference. Draw a circle around each pair of steps that can be run in parallel.



2f. Part (2e) assumes that multiple processors exist to perform certain steps of the algorithm in parallel. In general, multiple processes share a single processor giving rise to the need to coordinate the actions of individual processes. For example, a condition to be avoided is _____, which is the condition when two or more processes are all waiting for some shared resource that other processes of the group hold, causing all processes to wait forever without proceeding.

3. (10 pts)

3a. Suppose we want to perform binary search on a sorted array of elements, returning either a position in the array, or nil if the requested item does not appear in the array. Complete the following code:

```
def search(array,item)

  low = _____

  high = _____

  while _____ do

    mid = _____

    if array[mid] == item then

      return mid

    elsif array[mid] < item then

      low = _____

    else

      high = _____

    end

  end

  return _____

end
```

3b. Trace the binary search that results from the following function call (using the search function from problem 3a) by filling in the table below. You might not need to fill in all the rows.

`search([2,15,29,34,48,51,67,79,85,93], 67)`

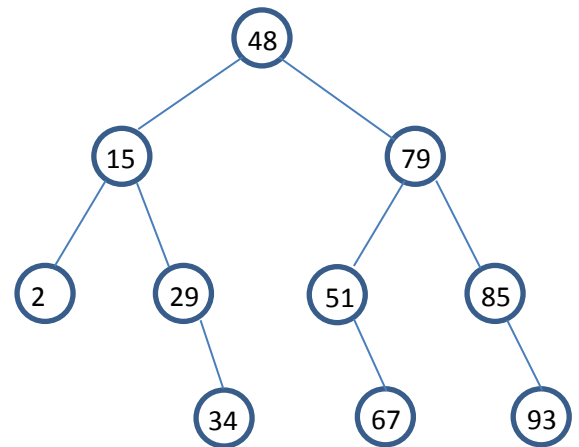
low	high	mid	array[mid]

(3c) Now consider doing binary search using a binary tree such as the one below. Note that each node is a list of the form `[item, left_child, right_child]`, where `item` is an integer and `left_child` and `right_child` are either nodes or `nil` (if that child is missing).

```
[48, [15, [2, nil nil],
        [29, nil,
          [34, nil, nil]]],
 [79, [51, nil,
        [67, nil, nil]],
      [85, nil,
        [93, nil, nil]]]]
```

Fill in the code for the binary search algorithm that operates on trees of this form:

```
def search(tree, item)
  if tree == _____ then
    return false
  elsif tree[0] == _____ then
    return true
  elsif tree[0] > item then
    return _____
  else
    return search(tree[2], item)
  end
end
```



3d. Given a balanced binary tree with n terminal nodes, what is the largest number of recursive calls that could be made by the search function of problem 3c when the item does not appear in the tree? _____

3e. Fill in the blank by circling one of the choices below. For maximum efficiency, a hash function should distribute items _____ across the buckets.

- A. randomly
- B. uniformly
- C. sequentially
- D. logarithmically

4. (10 pts)

4a. Complete the recursive function `doubler(list)` that doubles every element of a list and returns the result. Example: `doubler([2,5,30])` returns `[4,10,60]`. Note: the “+” in the code below denotes array concatenation, not integer addition.

```
def doubler(list)
    if _____
        return _____
    else
        return [list[0]*2] + _____
    end
end
```

4b. Consider the recursive function `twos(x)` shown below, that factors all the 2’s out of an integer. Find the result of `twos(80)` by tracing each recursive call, by filling in the table below. Hint: begin by filling in the call column until you hit the base case, then work your way back up to fill in the result column.

```
def twos(x)
    if x%2 == 0 then
        return [2] + twos(x/2)
    else
        return [x]
    end
end
```

#	Call to twos	Result
1	<code>twos(80)</code>	
2		
3		
4		
5		

4c. What does the function `f(x,y)` shown below do? Hint: consider `f(3,4)`.

```
def f(x,y)
  if x == 0 then
    return y
  else
    return f(x-1, y+1)
  end
end
```

4d. A palindrome is a sequence that reads the same forward and backward, such as “racecar” or [1,4,7,4,1]. An empty sequence is obviously also a palindrome, as is a sequence of length one. In Ruby we can test for palindromes with `x==x.reverse`, but in this problem we’re going to write a recursive test, dropping both the first and last element of the sequence with each recursive call. Fill in the missing code. Note that this solution works for both lists and strings.

```
def palindrome?(x)

  if x.length _____ then

    _____

  elsif x[0] != x[x.length-1] then

    _____

  else

    _____

  end

end
```

5. (10 pts)

5a. In a reliable message delivery system messages are received by the recipient in the order they are sent by the sender. If we wanted to implement a message buffer that stores the messages in transmission which data structure would be the most appropriate: *graph, queue, stack* or *tree*? _____

5b. If you have an array of 5000 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? _____

5c. If you have a linked list of 5000 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? _____

5d. Consider the function `balanced?(list)` that returns *true* if its input is a properly balanced parenthetical expression, and *false* otherwise. An expression is properly balanced if every left parenthesis is matched by a closing right parenthesis, and every right parenthesis closes some left parenthesis. We can use a stack to keep track of all the

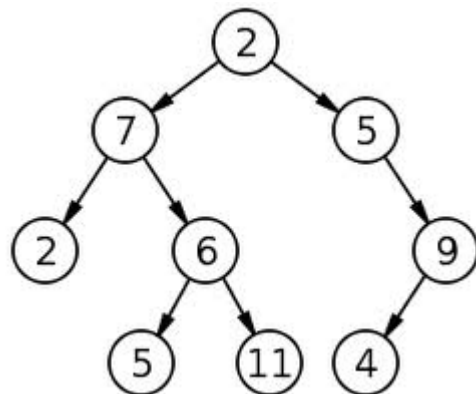
parentheses waiting to be closed. In a balanced expression, a right parenthesis will never be encountered when the stack is empty, and the stack will always be empty after we've processed the last element of the input. Examples: `balanced?(["(", "(", ")", ""])` should return *true*, and `balanced?(["(", ")", ")", "(", ""])` should return *false*. Complete the function.

```
def balanced?(list)
  stack = []
  for x in list do
    if x == "(" then
      stack.push(x)
    else # x must be ")"
      if stack.empty? then
        _____ # no left paren to close!
      else
        _____
      end
    end
  end
  return stack.empty?
end
```

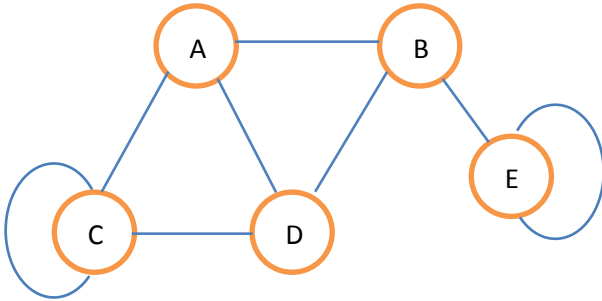
5e. Answer the following questions based on the tree given below.

(i) Is the given tree a binary tree? Why or why not? _____

(ii) Is the given tree a binary search tree? Why or why not? _____



5f. (i) Recall that an undirected graph can be represented using an adjacency matrix M such that for all i and j , $M[i][j] = 1$ and $M[j][i] = 1$ if there is a link between i and j , and $M[i][j] = 0$ otherwise. Given the example graph below, write down its 5-by-5 adjacency matrix.



(ii) If a given adjacency matrix M is such that $M[i][i] = 0$ for all i , can the graph have any self-links?

Yes or No? _____

6. (10 pts)

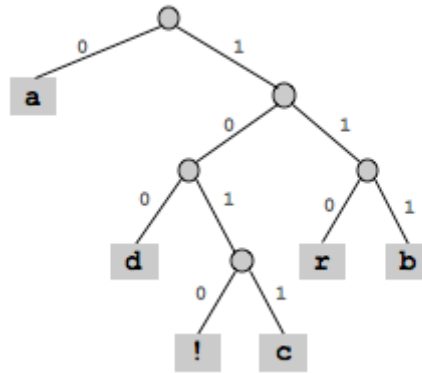
6a. Write the integer +73 as an 8-bit signed integer.

(The powers of 2 are: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...)

6b. Based on your answer from part (a), write the integer -73 as an 8-bit signed integer.

6c. How many distinct colors can be represented using the Red-Green-Blue (RGB) encoding for images? Note that in this encoding the intensity of each color component is represented using 8 bits. You may express your answer as a power of 2.

6d. A Huffman tree is designed for a collection of 6 characters as shown below.



(i) Based on this Huffman tree, we can conclude that the letter a is _____ commonly used than the letter c. Fill in the blank with the answer “less” or “more”.

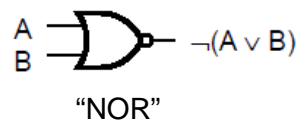
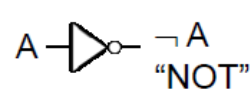
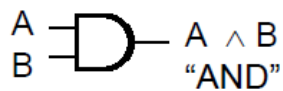
(ii) Decode the following message that was encoded using the given Huffman tree:

0111110010110100011111001010

6e. Complete the following truth table:

X	Y	Z	$(X \wedge Y)$	$(\neg Y \wedge \neg Z)$	$(X \wedge Y) \vee (\neg Y \wedge \neg Z)$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

6f. A gate is an electronic device that operates on a collection of binary inputs to produce a binary output. Below are diagrammatic representations for a set of logic gates.



Draw a diagram for the formula $(X \wedge Y) \vee (\neg Y \wedge \neg Z)$ using AND, OR, and NOT gates.

6g. Draw a diagram for the same formula from part (6e) this time using AND, OR, and NOR gates. Note that you will need to make a simple transformation using one of DeMorgan's laws on part of the expression.

6h. Put the following computer components in order from lowest to highest level of abstraction. Hint: If a component M is used to build a component N then M is at a lower abstraction level than N.

- A. Full adder
 - B. Transistors
 - C. Central Processing Unit
 - D. Gates
 - E. Arithmetic Logic Unit
-

7. (10 pts) Recall that Ruby has a `rand(n)` function that returns a random integer between 0 and n-1, inclusive.

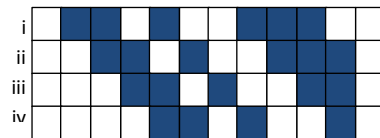
7a. Write a Ruby expression to generate a random integer value between -5 and 5, inclusive. Hint: how many possible values are there?

7b. Show how to use `rand` to generate a random string from the list below.

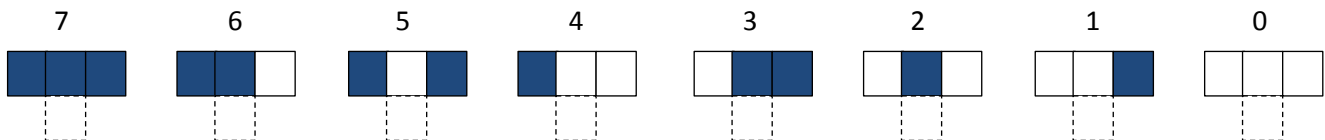
```
list = ["rock", "paper", "scissors"].
```

7c. In a continuous-time simulation, such as the oscillatory motion of a weight attached to a spring, time can't really be continuous. How do we deal with this?

7d. Consider a 1D cellular automaton where with each new generation, a cell takes on the value of its left neighbor. Thus, whatever pattern the automaton was initialized to would shift one place to the right with each new generation, and gradually be replaced by all zeros (white squares) shifting in from the left. Here is an example initial state and the next three generations. Note that the first and last cells are fixed at 0 (white). And if a square does not exist and is outside of the picture, then it is white.



Write down the rule for this automaton by putting 0's and 1's in the dotted boxes below.



7e. Write down the number of the rule you created in problem 7d. _____

8. (10 pts)

8a. Match the following:

- | | |
|------------------------------|--------------------------|
| _____ IP address | 1. http, https |
| _____ domain name | 2. RSA |
| _____ protocol | 3. www.cmu.edu |
| _____ URL | 4. Packet switching |
| _____ encryption | 5. 128.2.1.101 |
| _____ TCP/IP reference model | 6. Link layer |
| _____ router | 7. http://www.amazon.com |

8b. Given rows A, G, O, and T of the Vigenere table:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S

Decode the following message using the keyword GOAT :

M F A I K Z E T B S S _____

8c. What idea did Jacquard's loom contribute to computing? _____

8d. Fill in the blanks using the terms below; some may be used more than once, and others not at all. Alice wants to publish a document and allow any recipient to verify that she is the author and that the document is unmodified. To do this, she computes a _____ of the document and encrypts this with _____; we call this a _____. She then publishes the document along with her signature. When Bob reads this, he can verify the document's authenticity by decrypting the _____ using _____, then verify that it matches the _____ of the document.

- | | |
|------------------------|--------------------------|
| a. Alice's public key. | f. prime factorization |
| b. Alice's private key | g. digital signature |
| c. Bob's public key | h. ciphertext version |
| d. Bob's private key | i. plaintext version |
| e. MD5 hash | j. key exchange protocol |

8e. In a quantum computer with n independent qubits, the state space can be described by how many amplitude values? _____

When the qubits are fully entangled, then the state space has _____ amplitudes.

9. (10 pts)

9a. In the game of Capture, there is a board with 16 numbers in a 4 X 4 grid as shown below. Players alternate turns placing a stick on an empty horizontal or vertical line on the board. The player that places the fourth stick around a number captures that number. The player with the highest total captured at the end of the game wins.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Original game board

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Player 1 plays
between 9 and 10.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Player 2 plays
between 3 and 7.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Player 1 plays
below 16. etc...

If we build a game tree to analyze all of the moves of this game starting from an initial game with an empty board:

- (i) How many nodes would be at the first level below the root? _____
- (ii) How many nodes would be at the second level below the root? _____
- (iii) How many terminal nodes (nodes without any children) would there be? _____

9b. Suppose that you are given a pattern expressed as the regular expression `/ . a . * /` in Ruby. Write 2 strings that would yield a successful match against this pattern.

- (i) _____
- (ii) _____

9c. An engineer is designing a digital circuit with 10 separate input lines that are interconnected through some logic gates to output a single bit. What is the maximum number of input settings he would have to test in order to find out if the circuit would ever output the bit 1? _____

9d. If an algorithm has $O(n^4)$ running time do we classify it as tractable or intractable? _____

9e. P is the class of problems that can be decided in polynomial time and NP is the class of problems for which a candidate solution can be verified in polynomial time. Which one of the statements below is correct about P and NP. Circle your answer.

- i. $P = NP$
- ii. $P \neq NP$
- iii. Not proven whether i or ii.

9f. A fundamental result in computability theory, proved by Alan Turing in 1936, establishes that it is not possible to write a single universal program that can determine if any program _____ on any given input.

10. (10 pts – 5 pts each)

10a. Write a Ruby function `first_even(list)` that takes as argument a list consisting of integers and returns the position of the first element that is an even number if such an element exists, and returns `nil` if all elements are odd. You can assume that 0 is an even number.

10b. Write a Ruby function `max_rand(c, n)` that uses the `rand` function to print `c` random numbers in the range 0 to `n`, one per line, and return the maximum number that is generated. You can use a for loop for this; you will have to keep track of the maximum value seen so far as you iterate.

BONUS QUESTIONS (1 pt each) – Your answer must be exactly correct to receive the bonus point.

1. What is the name of the procedure that has been famously proposed to determine whether a computer has achieved “artificial intelligence”?

2. What is the value of this Ruby expression? Remember that Ruby evaluates expressions left to right, and pay careful attention to the nesting of delimiters; there is no typo here. `[[[[2]]][[].length].length]`
