15110 PRINCIPLES OF COMPUTING – EXAM 1A – FALL 2012

1	_____
2	_____
3	_____
4	_____
5	_____
6	_____
TOTAL	_____

Name _____ Section _____

Andrew id _____

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.(14 pts total) History of computation

(a) (4 pts) Match each item in the left column with the most relevant item in the right column.

Jacquard's loom	___6___	1. Electromechanical computer
ENIAC	___4__	2. Enigma cipher
Moore's Law	___7___	3. Polynomial function
Difference Engine	___3___	4. Vacuum tubes
Harvard Mark I	___1___	5. First programmer
Grace Hopper	___8___	6. Hollerith tabulating machine
Ada Lovelace	___5___	7. Exponential function
Alan Turing	____2__	8. Debugging

(b) (6 pts) We want to use the method of finite differences to create a table of numbers for the function $f(x) = 5x^2 - 4x + 3$. Compute the necessary difference functions and the initial values for x = 0 for this machine. Then fill in the table for x values from 1 to 3.

$\Delta f(x)$ = ____10_____ x + _____1____

$\Delta^2 f(x)$ = _____10_____

x	$\Delta^2 f(x)$	$\Delta f(x)$	f(x)
0	10	1	3
1	10	11	4
2	10	21	15
3	10	31	36

(c) (2 pts) A byte is ___8___ bits, so a kilobyte is ___2^{13}_____ bits.

(d) (2 pts) Moore's Law says that the _____number of (or complexity of)_____ of integrated circuit chips _____double_____ every 18 months.

2. This problem focuses on expressions and data types.

(a) (6 pts) For each of the following Ruby expression, write down the value that would be output if the expression was evaluated in irb.

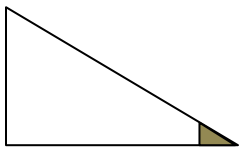
40 / 9	_____4_____	15.0 /2	_____7.5_____
2 * 2 ** 4	_____32_____	15 % 2	_____1_____
6 + 4 * 2 - 1	_____13_____	2 != 2	_____false_____

(b) (2 pts) Write a Ruby function `triangle_area` that takes two parameters `h` and `b`, respectively, for the height and base of a triangle, and returns the area of the triangle given by the formula

A = ½ (height × base).

```
def triangle_ area (h, b)
        return 0.5 * h * b
end
```

(c) (2 pts) Write a Ruby function `truncated_triangle` that takes height and base parameters (h and b) as input, and computes the area of a triangle that the tip cut off. The tip is also a triangle; its height and base are 10% of the height and base, respectively, of the larger triangle, as shown in the figure. Use the `triangle_area` function in your solution.



```
def truncated_triangle area (h, b)

    h_tip = 0.1 * h

    b_tip = 0.1  * b

   return   triangle_area(h, b) – triangle_area(h_tip, b_tip)

end
```

(d) (2 pts)

```
def mystery1(m, n)
     i = 0
     while i <= n-1 do
       i = i + 1
       print i ** m, " "
     end
   end
```

The Ruby function above prints a sequence of numbers. Which of the following is the output of the function expressed in terms of m and n? Circle your answer.

1 2^m 3^m … n^m OR 1 m^2 m^3 … m^n

The first one

(e) (4 pts) If the print statement was taken outside of the while loop to occur right after the while statement, as shown below, what would the function call mystery2(2, 3) print?

```
def mystery2(m, n)
  i = 0
  while i <= (n-1) do
    i = i + 1
  end
  print i ** m, " "
end
```

_____9_____

(f) (4 pts) Consider the following Ruby function:

```
def mystery3(m, n)
  i = 0
  result = 0
  while i <= (n-1) do
    i = i + 1
    result = result + i ** m
  end
  return result
end
```

What would the value of the variable x be after executing the following assignment statement below?

```
x = mystery3(2,4)
```

_____2 + 4 +8+16 = 30_____

(3) (20 pts total) This question focuses on the array data type and iterators.

(a) (6 pts) Assume the following list definition in Ruby using an array.

cars = [“Honda”, “Toyota”, ”Kia”, “Chrysler”, “Mercedes”]

What would be displayed in `irb` for each of the following Ruby expressions?

cars.length _____5_____

cars.first _____”Honda”_____

cars[1] _____”Toyota”_____

cars.include?(“Mazda”) _____false_____

cars.include?(“KIA”) _____false_____

_____Toyota*Mercedes*_____

(b) (4 pts) Assume the following list definition in Ruby using an array.

a= [1, 2, [3, 4, 5], 6]

What would be displayed in irb for each of the following Ruby expressions?

a.length _____4_____

a.first _____1_____

a[2] _____[3,4,5]_____

a + a _____[1,2,[3,4,5],6,1,2,[3,4,5],6]_____

(c) (10 pts) Assume the following list definition in Ruby using an array.

a = [2, 4, 6, 7, 8]

What would be returned in irb for if the following Ruby expressions are executed in the given order?

a.collect{ |x| x / 2 } _____[1,2,3,3,4]_____

a _____[2, 4,6, 7, 8]_____

a.select{|x| x.even?} _____[2, 4, 6, 8]_____

a.delete_if{ |x| x % 3 == 0 } _____[2, 4, 7, 8]_____

Hint: delete_if is a destructive method in Ruby.

a _____[2, 4, 7, 8]_____

4. (20 pts total) This question focuses on looping.

(a) (8 pts) We wish to define a Ruby function `out_of_order` that takes an "almost sorted" list as input and returns the first item that is not in ascending order. The function should return nil if the list is entirely in ascending order. For example, `out_of_order`([1, 5, 17, 12, 24]) should return 12, since 12 is less than the preceding item, 17. Complete the following iterative function `out_of_order`.

```
def out_of_order(list)
  index = __0_____
  while index < _____list.length-1___
    if ___list[index]_____________ > _____list[index+1]_____________
then
      return _____list[index+1]_______
    end
  index = ___index + 1_________
  end
  return ___nil______
end
```

(b) (8 pts) Consider the following recursive algorithm for returning the first item in a list that is not in ascending order, else nil. Complete the recursive definition of `out_of_order`.

1. If the list has fewer than two elements, return nil.
2. If the first element in the list is greater than the second element, return the first element.
3. Otherwise return the result of a recursive call on the tail of the list (i.e., everything beyond the first element.)

```
def out_of_order(list)
  if ____list.length_____ < 2 then
    return ____nil____
  elsif ___list[0]_________ > __list[1]_________ then
    return ____list[0]_______
  else
    return ____out_of_order(list[1..list.legth-1]_____________________
end
```

(c) (2 pts) Give an example of a six element list that would be a worst case input for `out_of_order`.

_____Any six element list with all elements in nondescending order_____

(d) (2 pts) What is the big O worst case complexity of `out_of_order`?

_____O(n)_____

5. (20 pts) This question deals with searching and sorting.

(a) (2 pts)

What is the big O complexity of binary search? _____O(log n)_____

What is the big O complexity of insertion sort? _____O(n^2)_____

(b) (6 pts) Fill in the table below to show how binary search would locate the value "e" in the list `["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k"]`. Use the binary search algorithm taught in the book and covered in lecture. Note: this table may contain extra rows.

Iteration	Low	High	Mid	list[mid]

1	-1	11	5	f
2	-1	5	2	c
3	2	5	3	d
4	3	5	4	e

(c) (6 pts) For each sorting algorithm described below, give its correct name:

- For each input item, find its proper position in the result list and add it at that position.

  _____insertion sort_____

- For each position I in the list, find the index of the smallest item at or to the right of position I, and swap `list[i]` with `list[index_of_smallest]`.

  _____selection sort_____

- Organize the inputs into N groups of size 1. Systematically combine adjacent groups to form N/2 sorted groups, each of size 2. Repeat the process, combining adjacent groups of size 2 to form N/4 sorted groups of size 4. Keep going until you have one sorted group of size N.

  _____merge sort_____

(d) (6 pts) Suppose we want to know if all the elements of a list are the same. For example, `all_same([1, 1, 1, 99, 1, 1])` should return false, but `all_same(["f","f","f","f"])` should return true. Here are two solutions. Fill in the missing elements.

```
def all_same1(list)
  sorted_list = list.sort
  if sorted_list[0] == sorted_list[ __list.length -1_____ ] then
```

```
    return ___true____
  else
    return ____false___
  end
end


def all_same2(list)
  list.each { |x| return ___false____ if ___list[x]____ != list[0] }
  return ___true___
end
```

What is the big O complexity of all_same1? _____0(1)_if we could assume that sort is a constant time operation but it is not. The complexity is hidden in sort. The best runtime for any sort is O(n log n), O(n^2) would also be acceptable.

What is the big O complexity of all_same2? _____0(n)_____

6. (6 pts) This question is based on your readings from the book *Blown to Bits*.

When you print a report using a laser printer, can you assume that no one can tell who printed it? Give a yes/no answer followed by a one sentence explanation.

No. Many color printers secretly encode the printer serial number, date, and time on every page they print.