

# 15110 PRINCIPLES OF COMPUTING – EXAM 1B – FALL 2012

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

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	___6___
Babbage	___8___
Hopper	___2___
Turing	___7___
Moore	___4___
Engelbart	___3___
Pascal	___5___
Napier	___1___

1. bones
2. a moth
3. a mouse
4. Doubling in complexity
5. Leibniz
- 6.Hollerith
7. Enigma cipher
8. Lovelace

(b) (6 pts) We want to use the method of finite differences to create a table of numbers for the function  $f(x) = -7x^2 + 2x - 5$ . 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.

$$\Delta f(x) = \underline{-14} \underline{\hspace{1cm}} x + \underline{\hspace{1cm}} \underline{-5}$$

$$\Delta^2 f(x) = \underline{-14} \underline{\hspace{1cm}}$$

x	$\Delta^2 f(x)$	$\Delta f(x)$	f(x)
0	-14	-5	-5
1	-14	-19	-10
2	-14	-33	-29
3	-14	-47	-62

(c) (2 pts) 1KB of memory is  $2^{10}$  or  $10^3$  (decimal number) bytes.

(d) (2 pts) In 1982 a state of the art microprocessor had 100,000 transistors. According to Moore's Law, when would you expect to see microprocessors with 200,000 transistors? Compute the correct year between 1982 and 1997, and *explain your reasoning*.

\_\_\_\_\_ If we assume beginning of 1982 then middle of 1983, since the number of transistors 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  $A = \frac{1}{2} (\text{height} \times \text{base})$ .

```
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 has 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.

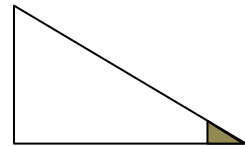
```
def truncated_triangle area (h, b)

  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<sup>m</sup> 3<sup>m</sup> ... n<sup>m</sup>

OR

1 m<sup>2</sup> m<sup>3</sup> ... m<sup>n</sup>

The first one

(e) (4 pts) If the print statement was taken outside of the while loop so that it occurred 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)
```

\_\_\_\_\_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?

<code>cars.length</code>	_____5_____
<code>cars.first</code>	_____ "Honda" _____
<code>cars[1]</code>	_____ "Toyota" _____
<code>cars.include?("Mazda")</code>	_____ false _____
<code>cars.include?("KIA")</code>	_____ false _____
<code>cars.each {  x  print x + "*" if x &gt; "Kia" }</code>	
_____ 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?

<code>a.length</code>	_____4_____
<code>a.first</code>	_____1_____
<code>a[2]</code>	_____ [3,4,5] _____
<code>a + a</code>	_____ [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?

<code>a.collect{  x  x / 2 }</code>	_____ [1,2,3,3,4] _____
<code>a</code>	_____ [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 `find_oddball` that takes an “almost sorted” list as input and returns the first item that is not in descending order. The function should return nil if the list is entirely in descending order. For example, `find_oddball (24,17,5,12,1)` should return 12, since 12 is greater than the preceding item, 5. Complete the following iterative function `find_oddball`.

```
def find_oddball (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 descending order, else nil. Complete the recursive definition of `find_oddball`.

1. If the list has fewer than two elements, return nil.
2. If the first element in the list is less 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 find_oddball (list)
  if __list.length____ < 2 then
    return __nil____
  elsif __list[0]____ < __list[1]____ then
    return __list[0]____
  else
    return __find_oddball(list[1..list.length-1]____)
  end
end
```

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

\_\_\_\_\_ Any six element list with all elements in non-ascending order  
\_\_\_\_\_

(d) (2 pts) What is the big O worst case complexity of find\_oddball?

\_\_\_\_\_ O(n) \_\_\_\_\_

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

(a) (2 pts)

What is the big O complexity of linear search? \_\_\_\_\_ O(n) \_\_\_\_\_

What is the big O complexity of merge sort? \_\_\_\_\_ O(n log n) \_\_\_\_\_

(b) (6 pts) Fill in the table below to show how binary search would locate the value "g" 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	5	11	8	i
3	5	8	6	9

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

- 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\_\_\_\_\_

- 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\_\_\_\_\_

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

\_\_\_\_\_insertion\_\_\_\_\_

(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])` returns false, but `all_same(["f", "f", "f", "f"])` should return true. Here are two solutions. Fill in the missing elements.

```
def no_mismatch1(list)
```

```
    list.each { |x| return ___false___ if ___list[x]___ != list[0] }
```



```

    return __true__
end

def no_mismatch2 (list)
    sorted_list = list.sort
    if sorted_list[0] == sorted_list[ __list.length-1_____ ]
then
        return __true__
    else
        return __false__
    end
end
end

```

What is the big O complexity of no\_mismatch1? \_\_\_\_\_  $O(n)$  \_\_\_\_\_

What is the big O complexity of no\_mismatch2? \_\_\_\_\_  $O(n \lg n)$  or  $O(n^2)$  \_depending on what is the complexity of sort \_\_\_\_\_

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

In parking garages, people with big parking bills sometimes attempt to get a second ticket at the end of their trip so that they would have to pay a smaller fee at the time of exit. What technology is used in many garages to prevent this? Give a one sentence answer.

Every car entering the garage has its license plate photographed at the same time the ticket is being taken.