

15-110 PRINCIPLES OF COMPUTING – LAB EXAM 1 – FALL 2012

A

Name _____ Section _____ Andrew ID _____ Machine _____

Directions:

1. In your home directory, create a folder named `labexam1`.
2. Write a function in Ruby for each of the following problems using `gedit` and store these functions in the `labexam1` folder. Test your functions by calling them within `irb`. Although we give you example /test runs, your function should work on all legal inputs based on the specifications given, and your output should match the examples as closely as possible for full credit. Remember that we will run your code on additional test cases that are not shown on the exam.
3. These problems can be done using for loops, while loops, each, or recursion: your choice.
4. Once you are finished, compress the `labexam1` folder into a zip file and submit it to AutoLab (<http://autolab.cs.cmu.edu>) by the end of lab. Do not delete the `labexam1` folder from your home directory.

Ruby syntax reminder:

```
def example1(x)
  for i in 0..x.length-1
    if x[i].odd? then
      puts x[i]
    end
  end
end
```

```
def example2(x)
  i = 0
  while i < x.length do
    if x[i].odd? then
      puts x[i]
    end
    i = i + 1
  end
end
```

1. (25 pts) Write a Ruby function `f1(x, n)` (in the file `f1.rb` in your `labexam1` folder) that prints all the elements of the list `x` that are greater than `n`. Your function should always return `nil` when it is done.

Sample usage:

```
>> f1([3,53,23,103,43,93], 50)
53
103
93
=> nil
```

2. (25 pts) Write a function `f2(list)` (in the file `f2.rb` in your `labexam1` folder) that takes a `list` as input and prints successively shorter prefixes of the list, starting with the list itself. Finally, your function should return the number of lines it printed. Remember to use `"p"` instead of `"print"` to print the lists, in order to display them in Ruby notation.

Sample usage:

```
>> f2(["do", "re", "mi", "fa"])
["do", "re", "mi", "fa"]
["do", "re", "mi"]
["do", "re"]
["do"]
[]
=> 5

>> f2([])
[]
=> 1
```

3. (25 pts) Write a function `f3(list)` (in the file `f3.rb` in your `labexam1` folder) that takes a list as input and returns a list containing the element immediately after the first occurrence of the string "this", and its square. If "this" does not appear in the list, or is the last element of the list, your function should return `nil`.

Sample usage:

```
>> f3([10, 20, 30, "this", 40, 50])
=> [40, 1600]

>> f3([20, 30, 40])
=> nil

>> f3([10, 20, "this", 30, 40, "this", 50, 60])
=> [30, 900]
```

4. (25 pts) Write a function `f4(n)` (in the file `f4.rb` in your `labexam` folder) that draws a hollow right triangle of size `n`, and returns `nil`. Hint: First figure out how the number of blank spaces in a row is related to the row number. It may also be helpful to recall that `"string"*n` in Ruby prints `n` copies of `"string"`.

Sample usage:

```
>> f4(6)
*
**
* *
*  *
*   *
*****
=> nil
```