# 15-110 PRINCIPLES OF COMPUTING – LAB EXAM 1 – SPRING 2013 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 <u>all</u> 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 points) Write a Ruby function `f1(list)` (in the file `f1.rb` in your `labexam1` folder) that returns the sum of all the numbers in the input `list` that are divisible by 3. Your function should return 0 if the input list is empty or there is no number in the list that is divisible by 3.

**Sample usage:**

```
>> f1([])
=> 0
```

```
>> f1([7, 11, 14, 19])
=> 0

>> f1([-6, -3, -21, -99])
=> -129

>> f1([43, 56, 2, 18, 95, 33])
=> 51
```

2. (25 points) Write a Ruby function `f2(list, start, finish)` (in the file `f2.rb` in your `labexam1` folder) that takes as input a list containing integers, and returns a new list consisting of the strings "even" and "odd" such that the following holds: the new list contains as many items as there are between `start` and `finish` (inclusive) and indicates whether the corresponding element in the input `list` is even or odd.

**Note:** You can assume that the function `f2` is always called with a non-empty list such that `0 ≤ start ≤ finish ≤ list.length-1.` Recall that the integer 0 is an even number.

**Sample usage:**

```
>> f2([5, 7, 22, 56], 0, 0)
=> ["odd"]

>> f2([7, 11, 14, 19], 0, 3)
=> ["odd", "odd" "even", "odd"]

>> f2([24, 5, 32, 46, 19], 3, 4)
=> ["even", "odd"]

>> f2([24, 5, 32, 46, 19], 4, 4)
=> ["odd"]
```

3. (25 points) Write a Ruby function `f3(list)` (in the file `f3.rb` in your `labexam1` folder) that takes a list as input and returns the index of the **last** occurrence of the string "food". If "food" does not exist in the list, your function should return nil.

**Sample usage:**

```
>> f3([])
=> nil

>> f3(["food"])
=> 0

>> f3(["hi", "bye", "fast", "food"])
=> 3

>> f3([4, 23, "food", "hunger", "hello", "fast", "food", 3, 5])
=> 6
```

4. (25 points) Write a Ruby function `f4(n)` (in the file `f4.rb` in your `labexam1` folder) that takes a positive integer `n` as input and prints the `n` by `n` multiplication table for integers `1` through `n`. Hint: The first row contains multiples of 1, the second row contains multiples of 2 etc. Your function should return nil. **Note:** The formatting of the table is not important as long as you have `n` rows each of which has `n` numbers that are one space apart.

**Sample usage:**

```
>> f4(1)
1
=> nil

>> f4(3)
1 2 3
2 4 6
3 6 9
=> nil

>> f4(4)
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
=> nil
```