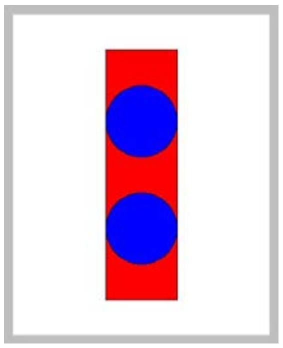
15110 PRINCIPLES OF COMPUTING – LAB EXAM 2 –FALL 2012 D

Name ___________________________ Section _____ Andrew ID ___________________ Machine_____

Directions:

1. *In your home directory (as soon as you start the terminal), create a folder named labexam2.*
2. *Write a function in Ruby for each of the following problems using gedit and store these functions in the labexam2 folder. Use the Ruby Reference Sheet to assist you with syntax issues.*
3. *Test your functions by calling them within irb. Although we give you sample test runs, your function should work completely based on the given specifications and your output should match the sample usage 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.*
4. *Once you are finished, compress the labexam2 folder into a zip file and submit it to AutoLab (http://autolab.cs.cmu.edu) by the end of lab. Do not delete the labexam2 folder from your home directory.*

1. (25 pts) Write a Ruby function `f1()` (in the file `f1.rb` in your `labexam2` folder) that draws two blue circles inscribed in a red rectangle, centered in a window of size 360 by 450. Note: the order in which you draw the shapes matters! Start with the rectangle. The circles should have radius 50 pixels and their centers should be 150 pixels apart. The rectangle should have width 100 pixels and height 350 pixels; its upper left corner is at x=130, y=50. The result should look just like the picture below, but if you're off by a pixel or two, it's okay. Refer to the Ruby Reference Sheet for help with Canvas method syntax.



2. (25 pts) Write a function f2(n) that generates an n×n identity matrix, i.e., a matrix with ones along the major diagonal and zeros everywhere else. You can assume n > 0. Use the following algorithm:

```
1. Create a variable result as a new array of length n.
2. For each row i from 0..n-1 of result:
    a. Set result[i] to a new array of length n.
    b. For each column j in row i, set that element to 1 if i equals j, else set it to 0.
3. When you've set all the elements, return result.
```

Sample usage:

```
>> f2(2)
=> [[1, 0], [0, 1]]

>> f2(3)
=> [[1,0,0],[0,1,0],[0,0,1]]

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

3. (25 pts) In the file `f3.rb` in your `labexam2` folder, write a recursive function `f3(list)` that returns all the elements of `list` that are divisible by 5. Do not use a for loop; you must use recursion.

Sample usage:

```
>> f3([2, 5, 75, 76])
=> [5, 75]

>> f3([20,50,52,55])
=> [20, 50, 55]

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

4. (25 pts) Write a function `f4(matrix)` (in the file `f4.rb` in your `labexam2` folder) that takes as input a matrix of numbers, and returns the product of all the elements. You may assume that each row of the matrix has the same number of elements in it.

Sample usage:

```
>> f4([[1, 2], [3, 4]])
=> 24

>> f4([[2,3],[5,7], [10, 0.5]])
=> 1050.0

>> f4([[]])
=> 1
```