Name : _____

Grading TA: _____

- INTEGRITY: By taking this exam, you pledge that this is your work and you have neither given nor received inappropriate help during the taking of this exam in compliance with the Academic Honor Code of Georgia Tech. Do NOT sign nor take this exam if you do not agree with the honor code.
- DEVICES: If your cell phone, pager, PDA, beeper, iPod, or similar item goes off during the exam, you will lose 10 points on this exam. Turn all such devices off and put them away now. You cannot have them on your desk.
- ACADEMIC MISCONDUCT: Academic misconduct will not be tolerated. You are to uphold the honor and integrity bestowed upon you by the Georgia Institute of Technology.
  - Keep your eyes on your own paper.
  - Do your best to prevent anyone else from seeing your work.
  - Do NOT communicate with anyone other than a proctor for ANY reason in ANY language in ANY manner.
  - Do NOT share ANYTHING during the exam. (This includes no sharing of pencils, paper, erasers).
  - Follow directions given by the proctor(s).
  - Stop all writing when told to stop. Failure to stop writing on this exam when told to do so is academic misconduct.
  - Do not use notes, books, calculators, etc during the exam.
- TIME: Don't get bogged down by any one question. If you get stuck, move on to the next problem and come back once you have completed all of the other problems. This exam has 8 questions on 7 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community. I have also read and understand the requirements outlined above.

Signature: _____

Question	Points	Score
1. Picture Mystery 2	4	
2. Vocabulary	9	
3. Lambda short answers	8	
4. Multiple Choice	2	
5. Replace A/E	10	
6. Short Answer	7	
7. Invert Color	12	
8. BigNumbersOnly	12	
Total:	64	

1. *(4 points)*

Examine the following code to determine what it does to the picture that is selected. Draw an original picture that is asymetrical (Label it "original"). Now, draw the picture that would result if the code below were ran on your original picture. (Label it "Result")

```
def Mystery():
   orig = makePicture( pickAFile() )
   result = copyPicture(orig)
   width = getWidth(orig)
   height = getHeight(orig)

   for x in range(0, width ):
      for y in range(0, height ):
         oPix = getPixel(orig, x,y)
         r,g,b = getRGB(oPix)
         rPix = getPixel(result, width-1-x, y)
         setRGB(rPix, (r,g,b))

   show(result)
```

Solution: This function will mirror the image horizontally.

Grading:
+4 clearly understands what the function does. (horizontal flip)
+2 Flips the image vertically, or both vertically & horizontally
+1 Rotates the image 90 degrees.

2. *(9 points)*
For each of the following vocabulary terms, write a concise 1-2 sentence definition. Be brief, and to the point.

(a) [3 pts] int

Solution: int - A numerical data type that stores numbers without a fractional component (such as -3, 5, 0, etc...)

(b) [3 pts] integer division

Solution: integer division - In python 2.x, when you divide two integers (int data type) the division will return an integer data type (int). Any remainder will be lost.

(c) [3 pts] local variable

Solution: A local variable is a variable that can only be seen (is visible) within the function that defined it. Once the function returns, the local variable is lost.

3. *(8 points)*
For each of the following lines, write what it returns when evaluated:

(a) [2 pts] `range(3,20,4)`

Solution: [3, 7, 11, 15, 19]

(b) [2 pts] `filter(lambda X: X > 7, [2,8,15,21] )`

Solution: [8,15,21]

(c) [2 pts] `map(lambda Y:  Y + 3, [1, 2,8,15] )`

Solution: [4, 5 ,11, 18]

(d) [2 pts] `reduce(lambda Z,W: Z+W, [2,4,16,20] )`

Solution: 42

4. *(2 points)*
For each of the following questions, select the appropriate answer by circling it.

(a) [1 pt] Order the following items from earliest (older) to latest (newer):

1. Konrad Zuse's Z1 computer

2. The Transistor
3. ARPANET

A. 1,2,3 B. 2,3,1 C. 1,3,2 D. 2,1,3 E. None of these.

(b) [1 pt] Order the following items from earliest (older) to latest (newer):

1. The Jacquard Loom
2. Ada Lovelace's program for the Analytical Engine
3. Jacques De Vaucanson's Digesting Duck

A. 1,2,3 **B. 3,1,2** C. 2,3,1 D. 2,1,3 E. None of these.

5. *(10 points)*

Write a function named `replaceAE` that takes in a string as a parameter. It should return a string where every 'a' in the original string is replaced with an 'e' and every 'e' is replaced with an 'a'. You may assume that all letters in the string are lowercase.

Example run:

```
>>> result =  replaceAE('andy dreams of flying rockets' )
>>> result
'endy draems of flying rockats'
```

Solution:

```
def replaceAE(inStr):
   outStr = ""
   for letter in inStr:
      if letter == "a":
         outStr = outStr + "e"
      elif letter == "e":
         outStr = outStr + "a"
      else:
         outStr = outStr + letter

   return outStr
```

Grading: 2 points for a correct header.
2 points for looping through each letter in the input string
2 points for correctly swapping A's and E's.
2 points for correctly building up the new string.
2 points for returning the correct new string.
-2 points if they swap any other letter accidentally. -4 if they leave out other letters.

6. *(7 points)*
For each of the following questions, give a brief answer:

(a) [5 pts] For each of the following algorithms, list their Big O complexity class:
Linear Search:
Binary Search:

Bubble Sort:
Insetion Sort:
Merge Sort:

Solution: Answer:

Linear Search: $O(N)$
Binary Search: $O(logN)$

Bubble Sort: $O(N^2)$
Insetion Sort: $O(N^2)$
Merge Sort: $O(NlogN)$
Grading: +1 for each correct line.

(b) [2 pts] Your intern has written an algorithm that will sort numbers. It can sort 1000 numbers in 1 second, and 5000 numbers in 25 seconds. What is the computational complexity (Big O) class of his algorithm?

Solution: Answer:
The algorithm has a computational complexity of N^2 (as N increases, the amount of time/work increases by the square.) This is equivalent to Bubble Sort or Insertion Sort. Grading: +2 for the correct answer. + 1 if they identify that it is similar to bubble sort or insertion sort but don't say N^2.

7. *(12 points)*

Write a function named `invert` that accepts a string parameter named fileName that contains the file name of a picture on disk. Your function must load the picture and invert the color of every pixel. To invert the color of a pixel, take each of it's color values and subtract them from 255. (i.e. newBlue = 255-originalBlue). Your function should return the newly inverted picture.

Solution:

```
def invert(fileName):
   pic = loadPicture(fileName)
   for pix in getPixels(pic):
      setRed(pix,255-getRed(pix))
      setGreen(pix,255-getGreen(pix))
      setBlue(pix,255-getBlue(pix))
   return pic
```

Grading: 1 points for correct header.
2 points for loading the picture from disk.
2 points for looping through all pixels.
3 points for getting the original R,G,B values
3 points for correctly inverting each value and putting back in the image .
1 point for returning the image.

8. *(12 points)*

Write a function named `bigNumbersOnly` that accepts the name of a file to open as a string parameter. The function should open the file, which will be formatted as follows (one number per line):

```
54
4.0
23.3
765.2
54
876.8
34
238
45
50
```

Your function must read in the file and return a *list* of all the numbers greater than 50 (numbers should be stored as floats). Remember to close your file after you are finished reading it. If you find duplicate numbers greater than 50, DO NOT include duplicates in the list!

Solution:

```
def bigNumbersOnly( filename ):
  f = open(filename, "r")  #"r" is optional, but encouraged!
  lines = f.readlines()
  f.close()

  myList = []
  for line in lines:
     num = float(line)
     if num > 50 and (not (num in myList)): #Also works: if num > 50 and num not in
         myList.append(num)
  return myList
```

Grading: 1 point for correct header, 1 point for opening file, 1 point for not using "write" mode. 3 points for reading everything from the file (one read, or readline() in a loop, or readlines(). 1 point for closing the file. 2 points for converting the numbers to float. 1 point for testing against 50, 1 point for checking for duplicates, 1 point for returning the list.