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 7 questions on 9 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. Vocabulary	3	
2. Read Functionals	5	
3. Dictionaries & Tuples	10	
4. Picture Drawing	10	
5. ColorBlind	10	
6. BigNumbersOnly	12	
7. Draw Border	14	
Total:	64	

1. *(3 points)*
   For the following vocabulary term, write a concise 1-2 sentence definition. Be brief, and to the point.
   (a) [3 pts] slice

   Solution: A subsequence copied from a sequence specified by a range of indices. The slice operator is: sequence[start:stop].

2. *(5 points)*
   Examine the following code and pretend you are the python interpreter.

```
myList = range(2,21,3)
res1 =  map(lambda x:x+5, myList)
res2 = filter(lambda x:x%2==0 or x%5==0, res1)
res3 = reduce(lambda x,y: x+y, res2)
```

   After this code has executed, write what the following variables point to:

   myList : ______________________________
   res1 : ______________________________
   res2 : ______________________________
   res3 : ______________________________

   Solution: (1 pt) myList -¿ [2, 5, 8, 11, 14, 17, 20]
   (1 pt) res1 -¿ [7, 10, 13, 16, 19, 22, 25]
   (1 pt) res2 -¿ [10, 16, 22, 25]

(1 pt) res3 -¿ 73
+1 bonus pt if they get it all correct.

3. *(10 points)*
Read the following pieces of code. Determine whether they will be executed successfully or not. If yes, show the result that would be printed. If not, explain why not.

(a) [2 pts]
```
myTuple1 = (1,2,3)
myTuple1.append(4)
print myTuple1
```

Solution: No. Because tuple is immutable, so there is not an append function for tuple.

(b) [2 pts]
```
myTuple2 = (1,2,3)
myTuple3 = (4)
print myTuple2+myTuple3
```

Solution: No. Because myTuple3 is not a tuple but an int.

(c) [2 pts]
```
myList=[0,3,4,1]
myList.remove(3)
print myList
```

Solution: Answer: Yes. [0, 4, 1]

(d) [2 pts]
```
myDictionary1={ a:1, b:2, c:3}
myDictionary1[d]=4
print myDictionary1
```

Solution: No. Because a, b, c, d here are neither characters nor strings. They will be treated as undefined variable names.

(e) [2 pts]
```
myDictionary2={ 10 : 1, 20 : 2, 30 : 3}
print myDictionary2[1]
```

Solution: Answer: No. Because elements in dictionary can only be accessed by keys but not indexes. In myDictionary2, 1 is not one of the keys.

4. *(10 points)*
Examine the following code. Draw a sketch of exactly what would be shown on-screen if the `draw` function were called.

```
from myro import *

def draw():
   pic = makePicture(100, 100)
   for x in range(100):
    for y in range(0, 100):
      pix = getPixel(pic, x, y)
      if x == y:
        setRed(pix, 0)
        setGreen(pix, 0)
        setBlue(pix, 0)
      if x > 80 and y < 20:
        setRed(pix, 0)
        setGreen(pix, 0)
        setBlue(pix, 0)
  show(pic)
```

Solution: Grading:
+2 - Draws a line
+2 - Draws the line from a corner to another corner
+2 - Draws the line from the top left corner to the bottom right corner
+2 - Draws a box
+2 - Draws the box in the top right corner

5. *(10 points)*
Write a function named `colorBlind` that accepts a picture as a parameter. (A picture object, not a string or file location). Your function should copy the picture and modify the copy as follows: Switch the red and green values of each pixel. For example, if a pixel initially had RGB values (255,30,72), your function should change them to (30,255,72). Return the modified picture.

Solution:

```
def colorBlind(image):
   image2 = copyPicture(image)
   for pix in getPixels(image2):
      red = getRed(pix)
      setRed(pix,getGreen(pix))
      setGreen(pix,red)
   return image2
```

Grading:
1 points for function header
2 points for iterating over pixels
2 points for setting red and green to something
2 points for switching properly (storing one of the values beforehand)
1 points for changing a copy, not the original 1 point for returning something other than None
1 point for returning the modified picture

6. *(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.

7. *(14 points)*
Write a function called `border` that accepts a single picture as a parameter. Your function should return a copy of the picture with a five-pixel-wide black border drawn over the outside of the picture. (The returned picture size will be the same as the original picture, but the outside five pixels on all four sides will be changed to be black.) The original picture should not be modified. This function must work on pictures of any arbitrary size. (You may assume it is larger than 10x10 pixels.)

Solution:

```
def border(apic):
   apic = copyPicture(apic)
   h = getHeight(apic)
   w = getWidth(apic)
   for i in getPixels(apic):
      x = getX(i)
      y = getY(i)
      if(x < 5 or x >= w - 5 or y < 5 or y >= h - 5):
         setRGB(i,(0,0,0))
   return apic
```

Grading:
+1 correct function header
+1 for copying the picture
+2 for modifying the copy (not the orignal picture) +8 for drawing on each side (top,bottom,left,right) (2pt each) +1 for using correct color (black, 0,0,0)

+1 for returning a picture

This page intentionally left blank. You may use it as scratch paper. If you place an answer on this page, box it, label it clearly, and indicate clearly on the original problem page that your answer is on this page.