Name : ____________________

1. *(2 points)*
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 10 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
TA Name	2	
2. Vocabulary	9	
3. Renegade	4	
4. Multiple Choice	6	
5. ImageMatch	8	
6. Encode / Decode	10	
7. Pawnee Filter	10	
Total:	49	

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] operator

Solution: Operators are special tokens that represent computations like addition, multiplication and division.

(b) [3 pts] pixel

Solution: pixel - An image/picture (when stored in a computer) is made up of tiny elements called Pixels. Each pixel contains color information (Red, Green and Blue) values for a small point in the image, and is located at a specific X/Y location within the overall image.

Looking for: Small elements that make up an image. Contain color information.

(c) [3 pts] robot

Solution: Any mechanism that is guided by automated control.

3. *(4 points)*
Read the following short piece of code carefully, and write down exactly what would be printed when it was executed:

```
def whatDoesThisDo( a ):
   print( a * 3 )
   print( a[::-1] )
   print( a[:3]  )
```

```
    print( a[::3] )

whatDoesThisDo("renegade")
```

Solution:

```
renegaderenegaderenegade
edagener
ren
red
```

+1 point for each correct line.

4. *(6 points)*
For each of the following multiple choice questions, indicate the most correct answer by circling it!

(a) [1 pt] Which compound data structure is NOT a sequence?

A. string

B. tuple

C. list

D. dictionary

E. All of the above are sequences.

(b) [1 pt] Opening a file without specifying the second parameter opens it for:

A. reading B. writing C. overwriting D. appending E. None Of These

(c) [1 pt] Which line of the following code raises the first exception? The code has no typos. Assume "input.txt" is valid and readable, with multiple lines of text.

```
Line 1:   x = open("input.txt")
Line 2:   lines = x.readline()
Line 3:   for aVar in lines:
Line 4:       lines = lines.replace("\n","")
Line 5:   close.x()
```

A. Line 1 B. Line 2 C. Line 3 D. Line 4 **E. Line 5**

(d) [1 pt] In 1964, Douglas Engelbart demonstrated the worlds first:

A. Transistorized Computer **B. Mouse** C. Tape Drive D. Vacuum Tube E. Solid State Memory

(e) [1 pt] What would the a variable point at after the following code is executed?

```
try:
  a = 2
  if a % 2 == 0:
     a = a+3
  a = a / 0
  a = a * 2
except:
  a = a + 3
  a = a / 2
```

A. 6.5 **B. 4.0** C. 3.5 D. 3.0 E. 2.5 F. 2.0 G. 1.5

(f) [1 pt] Which of the following is true about the keys in a dictionary?

A. An integer can be a key.

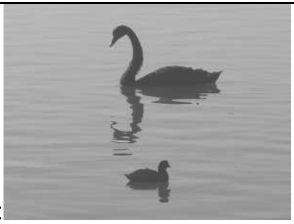
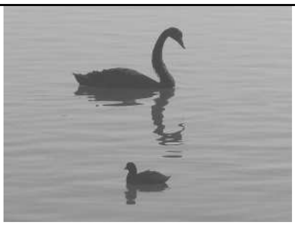
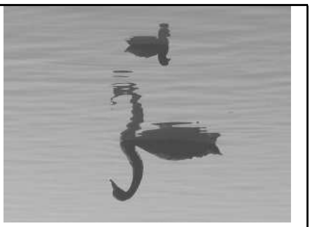
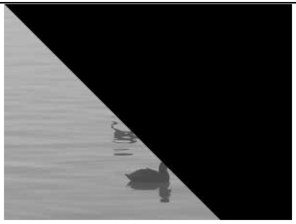
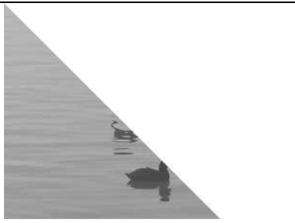
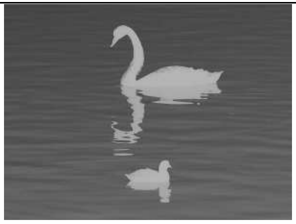
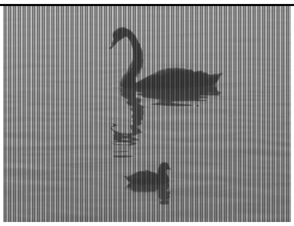
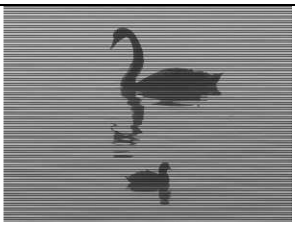
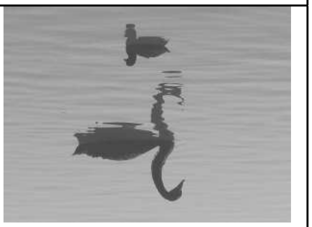
B. A list can be a key.

C. A dictionary can be a key.

D. All keys must be mutable.

E. All of the above are False.

5. *(8 points)*

Orig: 	A: 	B: 
C: 	D: 	E: 
F: 	G: 	H: 

The original swan image is in the top left of the table. For each boxed piece of code below, indicate which image it created by placing the image's corresponding letter inside the box. (Some images are not used.)

```
p=makePicture("swan.jpg")
p2 = makePicture(p)
h = getHeight(p)
for x in range( getWidth(p) ):
  for y in range( h ):
      pix = getPixel(p,x,y)
      pix2 = getPixel(p2, x,h-y-1)
      setRGB(pix2, getRGB(pix))
show(p2)
```

```
p = makePicture("swan.jpg")
for pix in getPixels(p):
   x = getX(pix)
   y = getY(pix)
   if x > y:
      setRed(pix,255)
      setBlue(pix,255)
      setGreen(pix,255)
show(p)
```

```
p = makePicture("swan.jpg")
for pix in getPixels(p):
   r = getRed(pix)
   g = getGreen(pix)
   b = getBlue(pix)
   setRed(pix,255-r)
   setBlue(pix, 255-b)
   setGreen(pix,255-g)
show(p)
```

```
p=makePicture("swan.jpg")
p2 = makePicture(p)
w = getWidth(p)
for x in range( w ):
  for y in range( getHeight(p) ):
      pix = getPixel(p,x,y)
      pix2 = getPixel(p2,w-x-1,y)
      setRGB(pix2, getRGB(pix))
show(p2)
```

Solution:

B	D
E	A

Grading: 2 points for each correct answer.

6. *(10 points)*
The ord function takes in a character and returns its ascii value, represented by an integer (*ord(’a’) returns 97*). The chr function does the opposite and takes in an integer and return the character corresponding to that value (*chr(97) returns ’a’*).

Using these built in functions, write two functions, named `decode` and `encode`. The encode function will accept a single string parameter and *return* a list of integers, one for each character in the string.

The decode function will take in a list of numbers as it’s parameter and *return* a string made up of the characters represented by those numbers.

For example:

```
>>> csStr = "CS1301 rocks!"
>>> csList = encode(csStr)
>>> csList
[67, 83, 49, 51, 48, 49, 32, 114, 111, 99, 107, 115, 33]
>>> csString = decode(csList)
>>> csString
’CS1301 rocks!’
```

Solution:

```
def encode( secretMessage ):
   secretList = []
   for ch in secretMessage:
      num = ord(ch)
      secretList.append( num )
   return secretList

def decode( secretList ):
   secretMessage = ""
   for i in secretList:
      secretMessage =  secretMessage + chr(i)
   return secretMessage
```

Grading:
Encode:
1pt - Correct def statement
1pt - Correctly iterates through characters
1pt - correctly converts characters to numbers with ord()

1pt - Correctly appends number to list
1pt - returns the list

decode:
1pt - Correct def statement
1pt - Correctly iterates through numbers
1pt - correctly converts numbers to characters with chr()
1pt - Correctly appends character to string
1pt - returns the string

7. *(10 points)*
You are to write a function for a supervisor that wishes to ignore emails from his annoying co-workers (all co-worker emails end with "@pawnee.gov"). A full log of these emails are stored in a text file where each line includes the sender's email, a colon, and the subject line. Your function should be named emailFilter and should take in one parameter, the name of the file (as a string). It should write out the emails that are NOT from co-workers to an out file called "readme.txt".

Example: If the emails.txt file contains the following...

```
Leslie@pawnee.gov: Ramsett Park Fiasco
Leslie@pawnee.gov: Ramsett Park URGENT
mulligans@gmail.com: BOGO 50% Steak Coupon
Ben@pawnee.gov: Break-even Analysis Tables Update
Jerry@pawnee.gov: My Presentation Today
cozysbar@gmail.com: Duke Silver gig Friday Night
Chris@pawnee.gov: Interested in more yoga?
doubletimestudio@yahoo.com: Recording Reservation Time
```

After running `emailFilter("emails.txt")` the readme.txt file should contain:

```
mulligans@gmail.com: BOGO 50% Steak Coupon
cozysbar@gmail.com: Duke Silver gig this Friday Night
doubletimestudio@yahoo.com: Recording Reservation Time
```

Solution:

```
def emailFilter(fileName):
   fIn = open(fileName, 'r')
   fOut= open('readme.txt', 'w')
   lines = fIn.readlines()
   for line in lines:
       if not ("@pawnee.gov" in line):
                        fOut.write(line)
   fIn.close()
   fOut.close()

Grading:

(10 points)
+1      correct header
+1      opens file using input parameter for reading
```

```
+1      opens out file named "readme.txt" for writing
+1      reads all the data from input file
+2      correctly filters out co-worker emails
+2      writes correct lines to "readme.txt"
+1      closes reading file
+1      closes writing file
```