

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 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	9	
2. Base Conversions	5	
3. Fill a Dictionary	3	
4. Searching	8	
5. Mystery Picture Function	6	
6. Picture Drawing	10	
7. WriteLines	8	
8. copyHalf	10	
Total:	59	

## 1. (9 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]`.

(b) [3 pts] element

**Solution:** One of the values in a list (or other sequence), selected using the bracket (indexing) operator.

(c) [3 pts] immutable data type

**Solution:** A (compound) data type in which the elements can not be modified.

## 2. (5 points)

For each of the following base conversion questions, give a brief answer:

(a) [1 pt] Convert the decimal number 17 into binary:

**Solution:** 10001

(b) [1 pt] Convert the decimal number 218 into binary:

**Solution:** 11011010

- (c) [1 pt] Convert the hexadecimal number 2F into decimal:

**Solution:**  $2 * 16 + F (15) * 1 = 32 + 15 = 47$

- (d) [1 pt] Convert the binary number 10011111 into decimal:

**Solution:** 159

- (e) [1 pt] Convert the decimal number 9 into hexadecimal:

**Solution:** 9

3. (3 points)

Examine the code below. Write the contents of the dictionary after the code runs. You may either list a table of key/value pairs, or write what would be printed to the screen if you executed `print myD` at the python prompt.

```
myD = {}  
  
for i in range(5):  
    myD[i] = i*20
```

**Solution:**

Key Value

```
0 0  
1 20  
2 40  
3 60  
4 80
```

Grading:

3pts = table completely correct. (They can also write out a dictionary using python syntax if they get it correct...)

2 pts - Forget one table entry.

1 pt - Get just the keys or just the values.

4. (8 points)

(a) [2 pts] Here is a sequence of numbers: 1, 2, 3, 6, 8, 13, 15

If you were to perform a linear search on the above sequence, looking for the number '8', in what order does the computer search through the items in the sequence?

Write them below.

**Solution:** 1,2,3,6,8

(b) [2 pts] If you were to perform a binary search on the original sequence for the number 8, in what order does the computer search through the items in the sequence?

Write each item that is examined in the order that it is examined below.

**Solution:** 6, 13, 8

(c) [2 pts] What advantage does the binary search give us?

**Solution:** Two point answers: -By cutting the list in half each time, it saves a lot of time. -It is  $O(\log n)$ , which executes much faster than a linear search.

2 points if they state that it's faster or saves time combined with WHY (cutting in half). Or know that it's  $\log N$  and that means it goes faster. 1 point if they just say it's faster but not why...)

1 point answers: -It saves time -It's faster.

- (d) [2 pts] Why might we use a linear search instead of a binary search?

**Solution:** Two point answers: -Because the list of numbers is not sorted. - Because it would take longer to sort a list of unsorted numbers than to just do the search.

5. (6 points)

Describe in words what the following function does. You should explain how it modifies the input picture/pixels and what it does with it once the modifications are complete.

```
def mystFunc(pic):
    for pix in getPixels(pic):
        r = getRed(pix)
        g = getGreen(pix)
        b = getBlue(pix)
        yel = makeColor(255,255,0)
        wht = makeColor(255,255,255)
        if r == 0 and g == 0 and b == 0:
            setColor(pix, wht)
        elif r == 255 and g == 0 and b == 0:
            setColor(pix, yel)

    show(pic)
```

**Solution:**

Grading:

- 1 point for indicating that it iterates over all pixels.
- 2 points for indicating that it replaces red pixels with yellow pixels.
- 2 points for indicating that it replaces black pixels with white pixels.
- 1 point for indicating that the modified picture is shown at the end.

6. (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

7. (8 points)

Write a function called `writeLines` that accepts a single string parameter that is the name of a file to open for writing. Your function should then use `raw_input()` to ask the user for input (potentially multiple times). Each time, display a prompt to the user that looks like this: "Please enter a line to write, or type quit to stop."

If the user enters a line of text and presses enter, your function should write that line out to the text file (including a newline), and then prompt the user for more input. If the user enters the word "quit" (without quotes) your function MUST NOT write quit out to the file. Instead, your function should close the file and return.

**Solution:**

```
def writeLines(filename):
    outF = open( filename, "w")
    prompt = "Please enter a line to write, or type quit to stop."
    line = raw_input(prompt)
    while line != "quit":
        outF.write(line)
        line = raw_input(prompt)
    outF.close()
    return
```

Grading:

1 point for correct function header.

1 point for opening the file using input parameter

1 point for opening using "w" mode.

1 point for prompting the user correctly (all times).

1 point for writing the line out to the file.

1 point for working on multiple lines (iteration).

1 point for stopping when the user types 'quit' 1 point for closing the file.

8. (10 points)

Write a function named `copyHalf` that accepts two pictures (`picA` and `picB`) as parameters. It should copy the left half of `picA` to cover/replace the left half of `picB`. (In effect, making `picB` into a new picture that is a "split-screen", the left half from `picA`, and the right half unchanged.) You should assume that `picA` and `picB` are exactly the same size.

Some example functions that might be helpful for this question:

- `getPixel(picture, x, y)`
- `getWidth(picture)`
- `getHeight(picture)`
- `getRed(pixel)`
- `setRed(pixel, value)`

**Solution:**

```
def copyHalf(picA, picB):
    middle = getWidth(picA) / 2

    for x in range(middle):
        for y in range( getHeight(picA) ):
            pixelA = getPixel(picA,x,y)
            pixelB = getPixel(picB,x,y)
            setRed(pixelB, getRed(pixelA) )
            setGreen(pixelB, getGreen(pixelA) )
            setBlue( pixelB, getBlue(pixelA) )
```

**Grading:**

2 points for correct header with 2 images as input

1 point for calculating middle value.

2 points for accessing the same pixel location in `picA` and `picB`

1 point for getting the RGB values out of `picA`

1 point for setting the RGB values into `picB`

3 points for correctly iterating through (copying data from) the left half of the images.