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 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. Vocabulary	9	
2. MysteryLists	5	
3. Multiple Choice	5	
4. OnlyAs	10	
5. World Cup	4	
6. Create Usernames	12	
7. Seek Light	8	
Total:	53	

1. *(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] float

   > **Solution:** A Python data type which stores floating-point numbers. Unlike INT's, floating point numbers have a fractional component (value after the decimal place).

   (b) [3 pts] iteration

   > **Solution:** Repeated execution of a set of statements.

   (c) [3 pts] robot

   > **Solution:** Any mechanism that is guided by automated control.

2. *(5 points)*
   Imagine that you are the python interpreter and execute the following code. Next to the code, write down exactly what it prints. Draw a box around your answer.

```python
def mystery(aList):
    if (len(aList) == 0):
        return []
    else:
        mystery(aList[1:])
        print(aList[1:])
```

```
a = [4,7,9,[6,7]]
b = a[:]
b[3][1] = 0
mystery(a)
```

Solution:

Printed:
```
[]
[[6, 0]]
[9, [6, 0]]
[7, 9, [6, 0]]
```

Grading:
1 pt - For largest printout, e.g. [7,9,[6,X]] (Ignore 7 vs 0 issues)
1 pt - For the [9, [6,X]] immediately before/after the longest printout.
1 pt - Answer in correct order (empty list first, longest list last)
1 pt - Empty list printed at either top or bottom.
1 pt - for [6,0] in nested list in all places it appears.

3. *(5 points)*
   For each of the following multiple choice questions, indicate the most correct answer by circling it!

   (a) [1 pt] Which of these commands is **NOT** a valid robot function?
       A. beep(1,4400)   **B. showName()**   C. takePicture()   D. forward(1,1)
       E. turnLeft(1,2)

   (b) [1 pt] Convert 110111_2 to decimal (base 10):
       A. 29   B. 53   C. 54   **D. 55**   E. 118

   (c) [1 pt] Convert 92_{10} to binary (base 2):
       A. 01010100   B. 11011100   C. 01011110   D. 01011101   **E. 01011100**
       F. 11011101

   (d) [1 pt] The ___was a direct replacement for bulky vacuum tubes and was a smaller, more reliable and power saving part.
       A. microprocessor   **B. transistor**   C. integrated circuit   D. mercury delay line

   (e) [1 pt] Who is widely recoganized as the first programmer?
       A. Grace Hopper   B. Charles Babbage   C. Alan Turing   **D. Ada Lovelace**
       E. Douglas Englbart

4. *(10 points)*

The function below accepts a file name (as a string) and writes every line in that file that begins with a capital letter A to a new file named newFile.txt.

Fill in the blanks appropriately:

```
def onlyAs(filename):
   inF  = _____   ( _____ , _____ )
   outF = _____   ( _____ , _____ )

   lines = inF._____
   for line in lines:
      if _____:
          outF._____

   inF.close()
   outF.close()
```

Solution:

```
def onlyAs(filename):
   inF = open( filename, "r")
   outF = open( "newFile.txt", "w")

   lines = inF.readlines()
   for line in lines:
      if line[0] == "A":  #or line.startswith("A")
          outF.write(line)

   inF.close()
   outF.close()
```

Grading: 1 point for each correctly filled in blank, with the exception of the if conditional. They get +2 points if that one is correct, +1 if it is close to correct.

---

5. *(4 points)*

Pretend that you are the python interpreter. Execute the following code and write what it prints out beside the code. If you have any other work on the page, draw a box around your answer.

```python
worldCup = {"Messi":(4,10), "Ronaldo":(1,13), "Muller":(5,10),
    "Neymar":[6,11], "Rodriguez":(7,14), "Cuadrado":(1,4)}
goals = 99
attempts = 99
player1 = ""
player2 = ""
for k in worldCup:
    if worldCup[k][1] < goals:
        goals = worldCup[k][1]
        player1 = k
    if worldCup[k][0] < attempts:
        attempts = worldCup[k][0]
        player2 = k

print(player1, goals)
print(player2, attempts)
```

Solution:

```
Line1: Cuadrado 4
Line2: Ronaldo 1
```

Grading:
1 pt - Cuadrado on first line. 1 pt - 4 on first line. 1 pt - Ronaldo on 2nd line. 1 pt - 1 on 2nd line.

6. *(12 points)*

Write a function called `createUsernames` that accepts three lists as parameters. The first parameter will contain strings which contain first names. The second parameter will be a list of strings that contain last names. The third list will contain integers. The function should combine each item in the lists to create a username. A username is made up of the first letter of the first name, up to the first six letters of the last name, and the digit. Note that if a last name has less than six letters the entire last name will be used.

Your funciton should return a list of usernames in the same order as the input data. You may assume that all three lists are of the same length.

For example:

```
>>> results = createUsernames(["Glenn","Ikenna"], ["HollingsWorth","Omekam"],
      [7,3] )
>>> print( results )
['GHollin7', 'IOmekam3']
```

Solution:

```
def createUusername(aList, aList2, aList3):
    newList = []
    for index in range( len(aList) ):
        string = aList[index][0] + aList2[index][:6] + str(aList3[index])
        newList.append(string)
    return newList
```

Grading:
1pt - Correct def statement
2pts - uses only first initial of first name.
2pts - Uses up to 6 characters of last name.
(-1 if they only grab 5 characters instead of six...)
2pts - Converts digit to a string.
2pts - concatenates each item the resulting username string.
2pts - Adds each username to a list.
1pt - returns the list of usernames.

7. *(8 points)*

Write a function named `seekLight` that accepts an integer as a parameter. The integer corresponds to how long your function should run, in seconds, before returning.

While running, your function should get the value of the robots' center light sensor (using a `getLight('center')` call). If the value returned is smaller than 500, your robot should move forward at full speed for one second. If the light sensor value is larger than (or equal to) 500, the robot should beep at 800Hz for 1 second, and not move.

Make sure that, no matter what, your robot stops and the function returns when the time is up! You may assume that the Myro library is already loaded and that your robot is already connected (init has been called) to the computer.

When your function returns, it should return None.

Solution:

```
def seekLight(time):
  for t in timer(time):
     if getLight("center") >= 500:
        beep(1,800)
     else:
        forward(1,1)
```

Grading: 1 point for correct header
1 point for running N seconds
1 point for checking the getLight sensor each iteration
2 points for moving forward when triggered
2 points for beeping when not triggered
(-2 points (instead of -4) if they reverse the beep and move)
1 point for function returning (None) when the time is up.
Misc Penalties:
-1 if doesn't stop at end of function.
-1 mixing-up time and frequency parameters beep(800,1)