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 5 questions on 6 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. Multiple Choice	6	
3. New Friends	5	
4. Square	10	
5. Rank Days	10	
Total:	40	

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] argument

   (b) [3 pts] proprioception

   (c) [3 pts] sequence

2. *(6 points)*

   For each of the following multiple choice questions, indicate the best answer by circling it.

   (a) [1 pt] What value is referenced by the variable x after this code is executed?
   ```
   x = 1
   count = 0

   while count < 4:
     if x % 2 == 0:
        x = x + 1
     else:
        x = x - 1
     count = count + 1
   ```
   A. 3   B. 2   C. 1   D. 0   E. -1   F. -2

   (b) [1 pt] Convert $9_{\{16\}}$ to decimal (base 10):

   A. A   B. 9   C. 7   D. 1C   E. 42

(c) [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

(d) [1 pt] Pretend you are the python interpreter and the following code is executed. What happens?

```
myList = [0,10,-2]
for x in range(10,15):
   if x % 3 == 0:
      print("Yes")
   if myList[x//10] == 10:
      print("Second")
```

A. An exception is generated

B. "Second" is never printed.

C. "Second" is printed exactly twice.

D. "Second" is printed exactly four times.

E. "Second" is printed exactly five times.

F. "Yes" is printed exactly twice.

(e) [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

(f) [1 pt] A file named "potato.txt" has three complete lines as follows:

```
Apple
Banana
Cake
```

The following code is run:

```
f = open("potato.txt", "r")
for line in f.readlines():
   print( line )
f.close()
```

How many total lines are printed to the console?

A. 0   B. 1   C. 3   D. 4   E. 6

3. *(5 points)*

Assume you have a file **oldFriends.txt** with the following contents:

```
Zara, 24
Michael, 18
Ceci, 28
Laura, 30
Luis, 45
```

Examine the following code:

```python
def myFriends(oldfile, newfile):
    infile = open(oldfile, "r")
    friends = infile.readlines()
    infile.close()

    friends.sort()

    outfile = open(newfile, "w")
    for friend in friends:
        newFriend = friend.split(",")
        if int(newFriend[-1]) < 30:
            outfile.write(friend)

    outfile.close()

myFriends("oldFriends.txt", "newFriends.txt")
```

Write down the contents of the newFriends.txt after the code above is executed:

4. *(10 points)*

   Write a function named `square`. This function will cause your robot to move in a square. The function will accept one integer parameter, representing the length of one side of the square in inches.

   You may assume that your robot moves one inch per second (forward or backwards), and turns 360 degrees in a time of 2 seconds (when traveling at full speed). When the function finishes running, it should beep in celebration and return the total distance travelled by the robot, in inches (excluding turns, as the robot does not "travel" while rotating).

   You may assume that `from Myro import *` has already been executed, and that the correct `init(...)` call has already been made, so your robot is ready to execute myro robot movement functions.

5. *(10 points)*

Write a function named `rankDays` which takes in one parameter, aList. The list will be a list of tuples, with each tuple containing information in the format (Number Processed, Day of Week). The Day of Week will be a string (either "Monday", "Tuesday", "Wednesday", "Thursday", or "Friday"), and the Number Processed will be an integer. You should create and return a dictionary which has the day's "rank" (a number 1-5, with 1 corresponding to the day with the largest number of items processed) as a key and the corresponding day of the week as the value. The ranking criteria will be on the numbers of items processed each day. You may assume that no two days will share the same number of items processed (i.e. each day will have a uniquely different number of items processed).

Example test case:

```
>>> aList = [(456, "Monday"), (371, "Tuesday"), (2316, "Wednesday"),
(1371, "Thursday"), (17, "Friday")]
>>> myDict = rankDays(aList)
>>> print(myDict)
{1: 'Wednesday', 2: 'Thursday', 3: 'Monday', 4: 'Tuesday', 5: 'Friday'}
```