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 will result in a substantial grade penalty.
  - 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 4 questions on 8 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. Multiple Choice	4	
2. Regular Expressions	8	
3. Halloween GUI	13	
4. Exam Grades	12	
Total:	37	

1. *(4 points)*
For each of the following multiple choice questions, indicate the single most correct answer by circling it!

(a) [1 pt] Which of these regex patterns would fully match an e-mail addresses that meets the following requriements: 1) Username must begin with at least one upper or lowercase letter (and optionally may have more letters followed by optional digits) 2) Hostname consists of lower or upper case letters only, seperated from a 3 letter top level domain by a period.

A. `[a-z][A-Z]+[1,2,3,4,5,6,7,8,9,0]@[a-z].[a-z]*3`

B. `[A-Z]+\d+.[a-z]+.[a-z]+[a-z][a-z]`

C. `[a-z]+[1,2,3,4,5,6,7,8,9,0]*.[A-Z]+.[a-z,a-z,a-z]`

D. `[a-zA-Z]?[a-zA-Z]*\d*@[a-zA-Z]+\.[a-z]{3}`

(b) [1 pt] Which of the following regular expressions will match **AND** capture ONLY 8 digit hexidecimal numbers (digits may only be 0-9 or A-F)?

A. `((?:[0-9]|[A-F]){8})`

B. `([^G-H]{8})`

C. `[\d|\w]{8}`

D. `([0-9]|[A-F]){8}`

(c) [1 pt] Which of the following regular expressions will match valid telephone numbers? Examples of valid telephone numbers: 1-800-225-5288 and 770-303-1992.

A. `(?:[^-]+\-)+[0-9]{4}`

B. `[0-9]?\-?[0-9]{3,4}\-?`

C. `(?:\d\-)?(?:(?:[0-9]{3})-){2}\d{4}`

D. `(?:\d\-)?(?:\d{3,4}\-?){3}`

(d) [1 pt] What costume will Jay wear for Halloween?

A. Tin-man B. Scarecrow C. Dorthy D. Toto E. Lion

2. *(8 points)*
Examine the following code:

```
from re import findall

callNums = '''813.52    F26p  M68f
813.69    F27a  B45
811       F20b  T33
839.9903  F27as
831.15    E27Z  B61'''
a = findall("([0-9]{3}\.?(?:\d\d){2})", callNums)
b = findall("[FEBMT][14093][0-9][abf]?", callNums)

ISBNs = '''99921-58-10-7
9971-5-0210-0
960-425-059-0
80-902734-1-6
1-84356-028-3'''
c = findall("\d+\-\d\d\d-\d{0,3}(?:\-\d)?",ISBNs)
d = findall("\d{3,}.+0-7",ISBNs)
e = findall("[0-9]+-[0-9]+[^0-9\n][0-9]{3}.\d",ISBNs)
```

What do the variables a,b,c,d and e refer to once the above code has been executed?

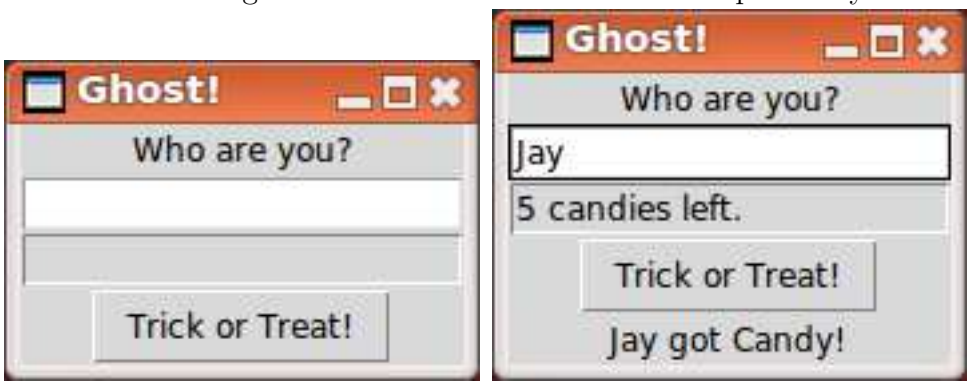
Solution:

```
a = ['839.9903']
b = ['B45', 'T33']
c = ['960-425-059-0', '84356-028-3']
d = ['99921-58-10-7']
e = ['960-425-059-0', '1-84356-028-3']

Grading:
 +1 point for each correct list element (total of 8)
 -1 for each extra/incorrect list element
```

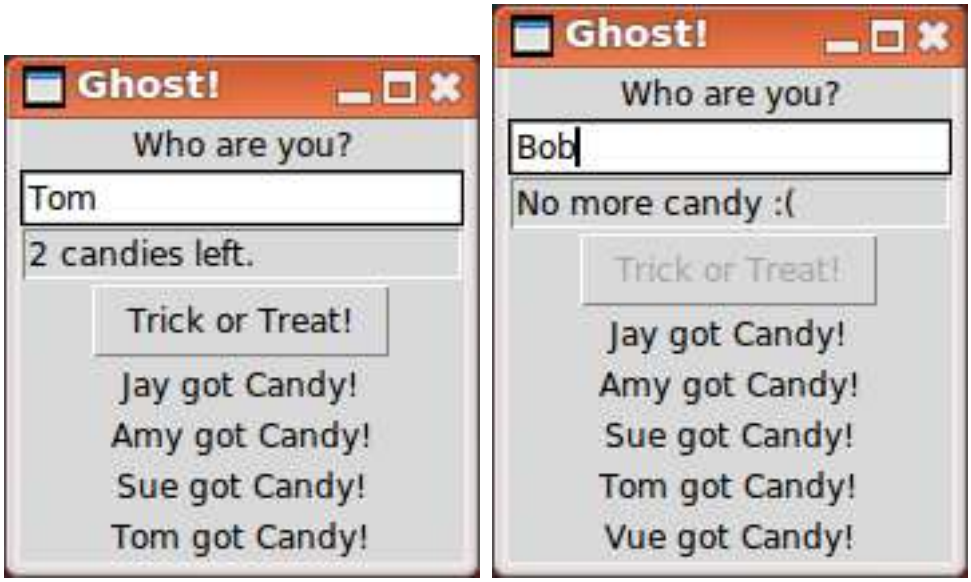
3. *(13 points)*

You are tasked with creating a GUI that looks like this to represent your door on Halloween.



You begin the night with 5 pieces of candy. Each time the "Trick or Treat!" button is clicked, you give one candy to the person at the door. (They have typed their name into the Entry widget below the "Who are you?" Label widget before the button is clicked.)

Inform the person at the door how many pieces of candy you have (before giving it to them, so the first person at your door will be told that you have 5 pieces).



Keep track of who has gotten candy by adding a Label widget to the bottom of your gui with their name each time the button is clicked. When you run out of candy, you must DISABLE your Trick or Treat button and change the message in the Entry box to say "No more candy :(".

Write python/tkinter code for this program on the next page. You may use classes, but are not required to do so. Don't forget to title your window appropriately!

Write your code here:

Solution:

```
from tkinter import *
class HalloweenGui:
    def __init__(self,root):
        self.root=root
        self.candies=5
        self.nameVar = StringVar()
        self.candyVar = StringVar()
        self.totButton = Button(root, text="Trick or Treat!", command=self.clicked)

        Label(root, text="Who are you?").pack()
        Entry(root, textvariable=self.nameVar,).pack()
        Entry(root, textvariable=self.candyVar, state="readonly").pack()
        self.totButton.pack()

    def clicked(self):
        if self.candies > 0:
            theStr = "{} got Candy!".format( self.nameVar.get() )
            Label(self.root, text=theStr).pack()
            theStr = "{} candies left.".format(self.candies)
            self.candyVar.set(theStr)
            self.candies = self.candies -1
        else:
            self.totButton.config(state=DISABLED)
            self.candyVar.set("No more candy :(")

win = Tk()
app = HalloweenGui(win)
win.title("Ghost!")
win.mainloop()


Grading:
+1 from tkinter import *
+1 window created with mainloop
+1 window title "Ghost!"
+1 Who are you label
+1 Name entry
```

```
+1 Output display entry
+1 Trick or Treat button
+1 decrements number of candies each time the button is pressed
+1 correct message ("{} candies left.")
+2 Adds Labels correctly each time the button is pressed
  (no points if pack and grid mixed in same container)
+2 disables Button and sets readonly Entry text correctly
   when there are no more candies left
```

4. *(12 points)*

A table has been created for you with the following command:
CREATE TABLE BOO (ID INTEGER NOT NULL UNIQUE AUTO_INCREMENT, NAME TEXT NOT NULL,PRICE FLOAT, RATING INTEGER, CATEGORY TEXT)

The database has contents such as the following (but with many more records):

ID	NAME	PRICE	RATING	CATEGORY
735	Bob the Builder	34.99	5	TV Show
866	Psycho Dorothy	NULL	NULL	NULL
398	Gothic Voodoo Doll	16.99	2	Creepy
849	Beetlejuice	19.99	4	Movie
354	Mummy	2.99	5	Minimal
666	Evil Jester	24.99	1	Creepy
423	Borat	13.23	5	Movie

Based on this table, write the SQL statements which will complete each task. Do not write any python code!

(a) [2 pts] Retrieve all of the data in the table ordered by the highest Rating first:

Solution: `SELECT * FROM BOO ORDER BY RATING DESC` Grading: +1 for select * from, +1 for order by rating desc

(b) [2 pts] Delete all records with a price greater than $18.00 (I'm on a college budget!):

Solution:
`DELETE FROM BOO WHERE PRICE > 18.00`
Grading: +1 DELETE FROM BOO, +1 for selecting where PRICE > 18.00

(c) [3 pts] Update "Psycho Dorothy" with a price of $9.99, a Rating of 3, and the category "Movie":

Solution: `UPDATE BOO SET PRICE=9.99, RATING=3, CATEGORY='Movie' WHERE ID = 866;`
Grading: +1 for update boo, +1 for price=, rating=, category =, +1 for WHERE...

(d) [2 pts] Retrieve (only) the name of the costume and the ID number from the BOO table where the cateogry is Movie:

Solution:
`SELECT NAME, ID FROM BOO WHERE CATEGORY = 'Movie'`
Grading: +1 for SELECT NAME,ID FROM BOO. +1 for WHERE CATEGORY='Movie'.

(e) [3 pts] Insert a new costume into the table. Set the name as the name of your costume, set the price to 9,999.99, Rating to 5, and the Category to 'Spontaneous':

Solution:

```
INSERT INTO BOO (NAME, PRICE, RATING, CATEGORY) VALUES
   ( <ANYTHING>, 9999.99, 5, 'Spontaneous')
```

Grading: +1 for INSERT INTO BOO +1 for (column labels), +1 for VALUES (...).

OK if they add an ID that is NOT in the example table of data.