CS 1301 – Spring 2009
Homework 5 – Scribby, go toward the light
Due: Friday, February 27at 6 PM
Out of 100 points

Files to submit: **1. hw5.py**
For Help:
- TA Helpdesk – Schedule posted on class website.
- Email TAs

Notes:
- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- **Do not wait until the last minute** to do this assignment in case you run into problems.
- If you find a significant error in the homework assignment, please let a TA know immediately.
- While HW 4 was a group assignment, this assignment should be done individually. Standard collaboration rules apply.

Part I – Finding the Lamp (50 pts)

Were you aware your robot has light sensors? Using a function called **getLight()**, you can determine on which side of the robot (left/right/center) there is the brightest light. More information on the getLight() function can be found here:

http://wiki.roboteducation.org/Chapter_5

This link includes some sample output of the getLight() function.

Your mission:

You robot will be placed in the vicinity of a light source. Your robot will need to act like a bug to a halogen lamp and go toward the light. [Remember that great part in A Bug's Life™ ? -- "Henry, don't look at the light!" "I can't help it, it's so beautiful!" ZAP!] Anyway, your robot is the bug in this case (though we won't be zapping Scribby). You will have **30 seconds** for your robot to locate the light and back into it. Backward, you say? Why yes. Remember, the getLight sensors are located on the BACK of the robot (thinking of the fluke as being the front), so keep this in mind when directing your robot on how to move while it's detecting light values. You robot should beep happily once the light is located (perhaps to celebrate getting the light without getting zapped.) [You can use the IR sensors to detect when you are close enough to the light to stop backing

towards it.] Please store your code in a function called **findLight()** – you will lose points for any other names. You are welcome to use helper functions as you find fit, but the findLight() should be the driving function behind your program.

Grading for Part I:

Hits the light within 30 seconds	**– 25 pts**
Uses getLight() to get values	**– 10 pts**
Adjusts movement based on light values	**– 10pts**
Beeps once objective is achieved	**– 5pts**
Total: 50 pts	

Part II - Silly Strings

For this part of the homework, you will be writing some functions to play around with strings. Some information on strings, as well as many methods you can use with them can be found here:

http://www.python.org/doc/2.4.4/lib/module-string.html

Part IIa – I Palindrome I (Pat on the back if you get the reference) – (15 pts)

Write a function called **isPalindrome()** which takes in a string and determines if the string is a palindrome. Your function should **return** either True of False (note the capitalization, i.e., you should be returning a Boolean, NOT the string "true" or "false"). Note: the case of the letters doesn't matter, it's the letter itself that does. See the sample output.

Sample Output:

```
>>> palindrome("kayak")
True
>>> palindrome("Banana")
False
>>> palindrome("a")
True
>>> palindrome("racecar")
True
>>> palindrome("rAcECar")
True
```

Note that a string consisting of a single character is considered a palindrome (it is after all

the same backwards or frontwards.)

Grading for Part IIa:

       Works for all cases   – 9 pts
       Returns a Boolean   – 6 pts (3 for return, 3 for the result being a Boolean)
           Total: 15 pts

Part IIb – Safe For Web (15 pts)

As you may notice when you visit various websites (see our helpdesk page for example), when someone posts their email address, they will often replace the "@" and "." in email addresses to keep scanning spambots from recognizing the address and adding it to a spam list. And who really wants spam? So, you're going to help people out a bit. Write a function called **safeForWeb()** which takes in an email address (as a string) in the typical format – "myName134@someDomain.com", and **returns** a string of an email address safe for web publication. To do this, you should replace the @ with [at] and the . with [dot]. Remember, some email addresses have several periods in them, (like gth657n@mail.gatech.edu).

You may find the string functions **split** and **join** to be useful. Check the docs (linked above) for more info on these.

Sample output:
```
>>> safeForWeb("gth657n@mail.gatech.edu")
'gth657n[at]mail[dot]gatech[dot]edu'

>>> safeForWeb("bob1234@someDomain.com")
'bob1234[at]someDomain[dot]com'
```

Grading for Part IIb:

       Correctly replaces the "@"   – 5 pts
       Correctly replaces all of the "."   – 5pts
       Returns a safe string   – 5pts
           Total: 15 pts

Part IIc – The Replacements (20 pts)

For your final silly string function, you will write a function called **replaceChar()** which takes in 3 parameters, a string, the character you want to replace, and what you want to

replace it with. Your function should then **return** a new string, consisting of the old string, but with the specified character replaced by the desired one. If the specified character does not appear in the string, then your function should just return the original string.

```
>>> replaceChar("Kane & Melody", "&", "<3")
'Kane <3 Melody'
>>> replaceChar("This is Sparta", " ", "!")
'This!is!Sparta'
>>> replaceChar("I Have A Cookie", "z", ":D")
'I Have A Cookie'
>>>
```

Grading for Part IIc:

       Removes every old character   - **7 pts**
       Replaces each with the new string - **7 pts**
       Returns the result   - **6 pts**
            Total: 20 pts

Part III – Turn In

Once you're done, please submit ONE file, hw5.py to T-Square. If you submit each function in a separate file, you will lose 10 points! The assignment is due Friday, Feb 27th at 6pm. If you're late, but turn it in before 6pm on Monday, you will lose 10%. We will not accept submissions after Monday.

Remember – if there are errors in a part of the homework, you will lose 50% credit for that particular part. If your file fails to run at all, you will lose 50% credit for the entire homework. Please test your code thoroughly!

Written By Melody Nailor, Spring 2009.