Lab01 - Python review and Jupyter Notebook

Before you come to the lab

Read this document carefully to properly prepare for the lab and turn in your lab solution (i.e., your lab report as per the instructions presented here).

1. Read Sections 1.8-1.12 of the textbook.
2. Install Jupyter Notebook on your personal computer. Jupyter Notebook has already been installed in our lab computers.
3. Launch Jupyter Notebook, and display the Lab Assignment Report template provided in the Lab01 page. To experiement with Jupyter Notebook, include some text, simple code, and images in your report. You may find this online resource (https://realpython.com/jupyter-notebook-introduction/#getting-up-and-running-with-jupyter-notebook) useful.
4. Study how to create unit tests in Python https://realpython.com/python-testing/ (https://realpython.com/python-testing/)
5. Research how to plot graphs in Python and Jupyter Notebooks using `matplotlib`

Prelude: The monkey typist

The *infinte monkey theorem* states that a monkey hitting keys at random on a typewriter keyboard for an infinite amount of time will almost surely type a given text, such as the complete works of William Shakespeare.

The `monkeyTypist()` function uses two helper functions: `generate()` and `calcScore()`. The code and respective output below illustrates how these functions operate. (I have manually interrupeted the execution of `monkeytypist()` while it was producing the output you see below.)

In [4]:
```
# This is just to illustrate what generate and calcScore return
target = list("me t")
lenS = len(target)
alphabet = list("abcdefghijklmnopqrstuvwxyz ")
lenA = len(alphabet)

print("String   Score")
for i in range(10):
    str = generate(lenS, lenA, alphabet)
    print("%s     %d" % ("".join(str), calcScore(str, target)))
```

```
String   Score
tezl     25
 xfq     0
 hm      0
txve     0
xgdv     0
rxtm     0
upez     0
 ceq     0
ogb      0
zqks     0
```

In [5]:
```
# Repeatedly calls generate and score while 100% of the letters do not match the t
arget string.
# If the letters are not correct then it will generate a whole new string.
# To make it easier to follow the program's progress it prints out the best
# string generated so far and its score every 100000 number of tries.

def monkeyTypist():
    target = list("methinks it is like a weasel")
    lenS = len(target)
    alphabet = list("abcdefghijklmnopqrstuvwxyz ")
    lenA = len(alphabet)
    epoch = 0
    bestStr = generate(lenS, lenA, alphabet)
    bestScore = calcScore(bestStr, target)
    print("String                          Score")
    while bestScore < 100:
        if bestScore==100:
            break
        newStr = generate(lenS, lenA, alphabet)
        newScore = calcScore(newStr, target)
        if newScore > bestScore:
            bestScore = newScore
            bestStr = newStr
        if epoch==100000:
            print("%s   %d" % ("".join(bestStr), bestScore))
            epoch = 0
        epoch = epoch+1

monkeyTypist()
```

```
String                          Score
apnw nts ntqfsjsixztmdkfjgeu   25
apnw nts ntqfsjsixztmdkfjgeu   25
 qgn utszxq zsgjikeho bbwsab   28
 qgn utszxq zsgjikeho bbwsab   28
 qgn utszxq zsgjikeho bbwsab   28
jtwzw seknv it lwke a rhj td   32

---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-5-4664d8cc0b42> in <module>()
     26         epoch = epoch+1
     27
---> 28 monkeyTypist()

<ipython-input-5-4664d8cc0b42> in monkeyTypist()
     16         if bestScore==100:
     17             break
---> 18         newStr = generate(lenS, lenA, alphabet)
     19         newScore = calcScore(newStr, target)
     20         if newScore > bestScore:

<ipython-input-3-8cf6a85012eb> in generate(lenS, lenA, alphabet)
      4     res = []
      5     for i in range(lenS):
----> 6         res = res + list(alphabet[random.randrange(lenA)])
      7     return res
      8

KeyboardInterrupt:
```

Exercise

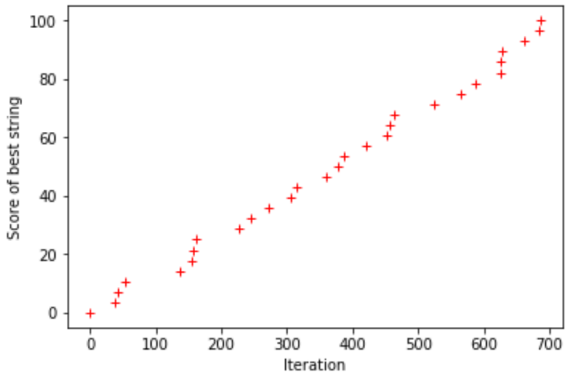
1. Create the following lab01 folder:

```
lab01/
│
├── mySolution/
│   └── __init__.py
├── test.py
├── lab01-report.ipynb
└── lab01-report.html
```

   - `__init__.py` should contain your python function definitions for this lab assignment
   - `test.py` should contain your tests.
   - `lab01-report.ipynb` should contain your jupyter notebook lab report.
   - `lab01-report.html` is the html version of your jupyter notebook lab report.
2. Implement your own `generate` and `calcScore` helper functions, and a better `monkeyTypist` function that keeps letters that are correct and only modifies one character in the best string so far. Your new `monkeyTypist` function should produce the following output:

   - print the best string and its score, every 100 tries, as seen in the output below.
   - plot a graph showing the scores of the best strings found and the iteration in which they were found. The x-axis should represent the iteration number, and the y-axis should represent the scores of the best strings as shown in the example below.

```
metxoaszyyqxoaayjqpseeq zkgz 10.714286 100
methinkayyqxoaayjqpseeq zkgz 25.000000 200
methinks iwxoaayjqpseeq zkgz 35.714286 300
methinks it is ojqpseeq zkgz 53.571429 400
methinks it is likeieeq zkgz 67.857143 500
methinks it is like a e zkgz 78.571429 600
methinks it is like a weasel 100.000000 687
```



3. In `test.py`, create test cases for helper functions `generate` and `calcScore`.
4. In your jupyter notebook Lab Report include a code cell that calls your function `monkeyTypist` to produce the desired output. For example, the code below when placed and executed in a Jupyter Notebook cell should produce an output similar to what you see above.

In [ ]:
```
from mySolution import monkeyTypist

monkeyTypist()
```

Preparing to submit your report

Create a zip file containing the following your `lab01` folder containing those folders and files mentioned in item 1 above.

What to submit

At the Lab01 web page in D2L, click on `Lab Solution Submission`, then attach and submit **only the zip** file you have created as per the instructions above.