**ACST 890- ASSIGNMENT 1 TASK 1 SOLUTIONS: 45535531-ROHAN MARCUS DEANS**

**GITHUB Username:** 45535531

**Repository Name:** Quiz1 (https://github.com/45535531/Quiz1.git)

**File Name:** 45535531DeansRohanMarcusA1T1.pdf

**Other Attached files on GITHUB:** codes.txt

**1.**

To convert all instances of the string 'Huck' to 'HucK', we first open the file using the following command:

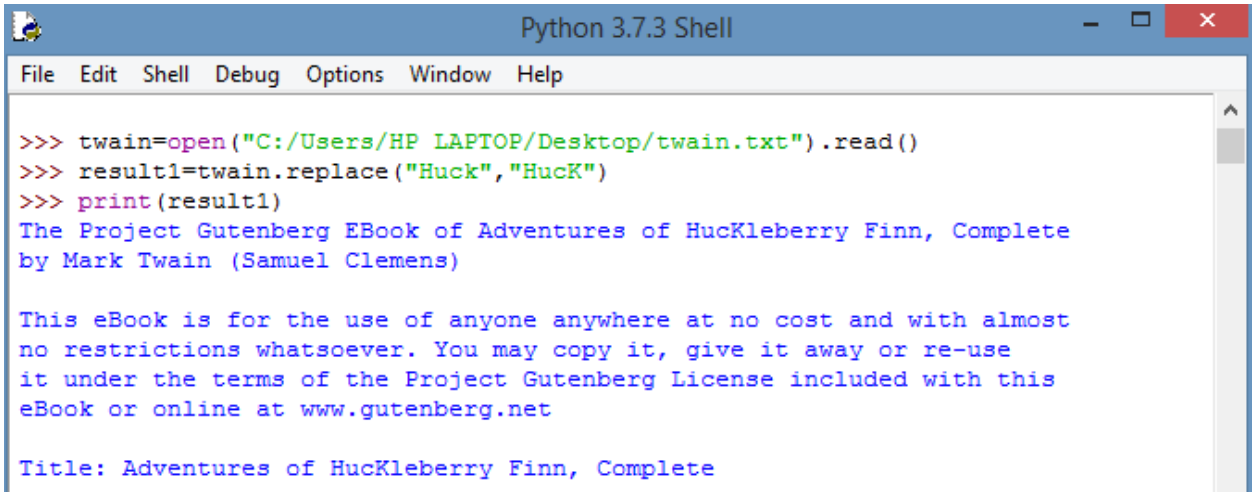
open("C:/Users/HP LAPTOP/Desktop/twain.txt").read()

To convert all instances of the string, we use the replace command as shown below:

```
>>> twain=open("C:/Users/HP LAPTOP/Desktop/twain.txt").read()
>>> result1=twain.replace("Huck","HucK")
>>> print(result1)
```

Squeezed text (12323 lines).

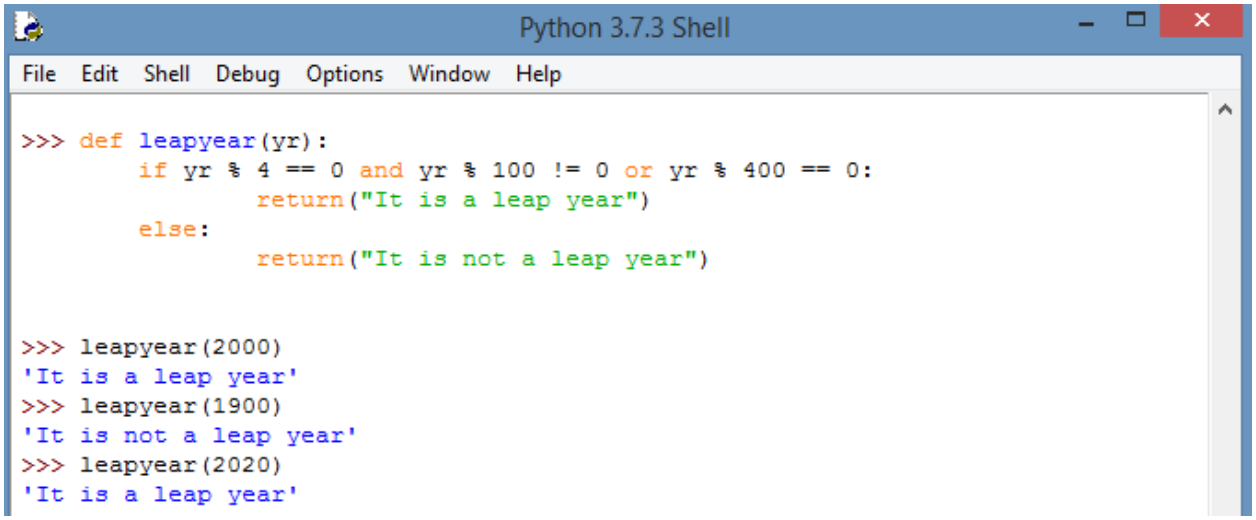
Since the file twain.txt is extremely large, I have included only a couple of excerpts of the output to show that all instances of Huck have been converted to HucK as shown below:




Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
CHAPTER I. Civilizing HucK.--Miss Watson.--Tom Sawyer Waits.
CHAPTER II. The Boys Escape Jim.--Torn Sawyer's Gang.--Deep-laid Plans.
CHAPTER III. A Good Going-over.--Grace Triumphant.--"One of Tom Sawyers's Lies".
CHAPTER IV. HucK and the Judge.--Superstition.
CHAPTER V. HucK's Father.--The Fond Parent.--Reform.
CHAPTER VI. He Went for Judge Thatcher.--HucK Decided to Leave.--Political Economy.--Thrashing Around.
CHAPTER VII. Laying for Him.--Locked in the Cabin.--Sinking the Body.--Resting.
CHAPTER VIII. Sleeping in the Woods.--Raising the Dead.--Exploring the Island.--Finding Jim.--Jim's Escape.--Signs.--Balum.
CHAPTER IX. The Cave.--The Floating House.
CHAPTER X. The Find.--Old Hank Bunker.--In Disguise.
CHAPTER XI. HucK and the Woman.--The Search.--Prevarication.--Going to Goshen.
CHAPTER XII. Slow Navigation.--Borrowing Things.--Boarding the Wreck.--The Plotters.--Hunting for the Boat.
CHAPTER XIII. Escaping from the Wreck.--The Watchman.--Sinking.
CHAPTER XIV. A General Good Time.--The Harem.--French.
CHAPTER XV. HucK Loses the Raft.--In the Fog.--HucK Finds the Raft.--Trash.

**3.**

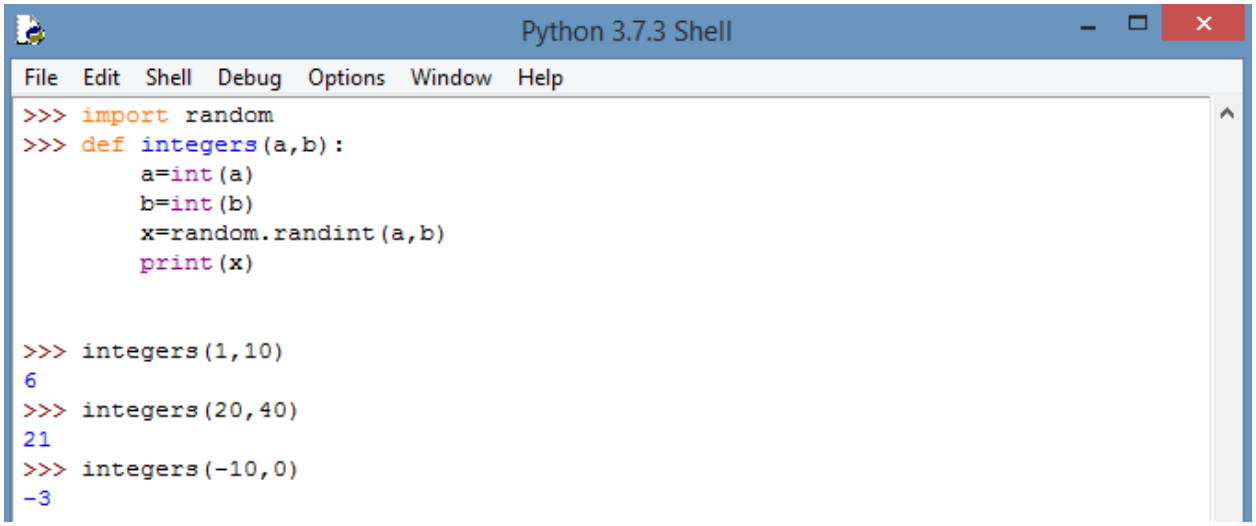
The following python code is used to test whether a given integer corresponds to a leap year in the Gregorian calendar as shown below:




To show the code executes as required, we use the following years: 2000, 1900 and 2020 as examples to the solution as shown above. Clearly 1900 is not a leap year, since it is not evenly divisible by 400 as compared to 2000 and 2020, which satisfy the conditions to be leap years.

**4.**

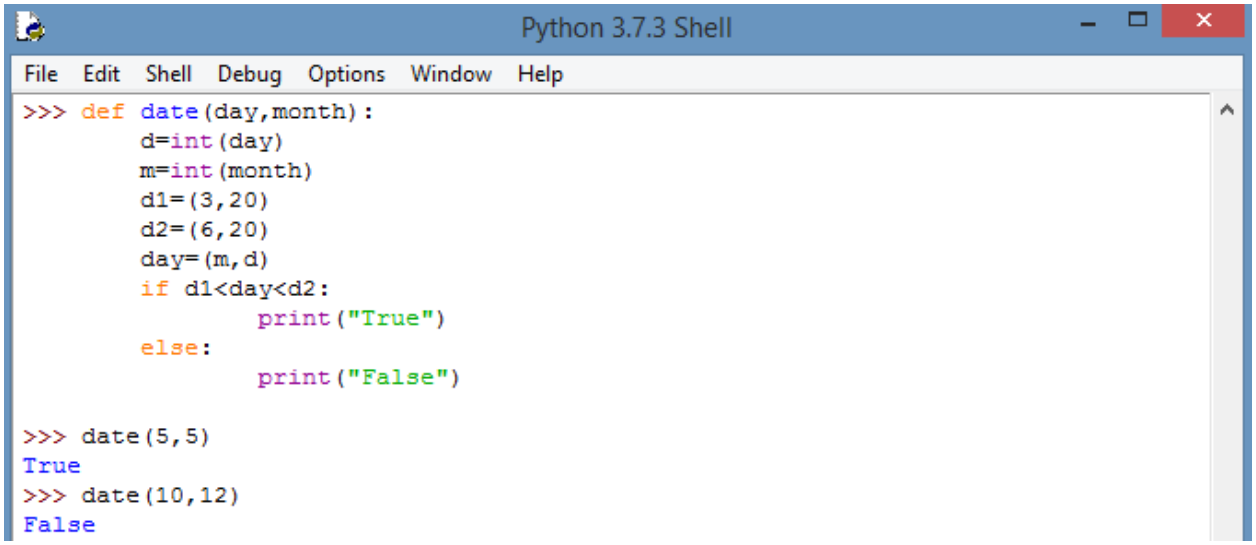
The following python program takes two integers a and b from the command line and writes a random integer between a and b. The command *import random* is used to implement the function *randint()*.




To show the code executes as required, we use the following ranges of integers: 1 and 10, 20 and 40, -10 and 0 between which we obtain a random integer in each case as examples to the solution as shown above.

**5.**

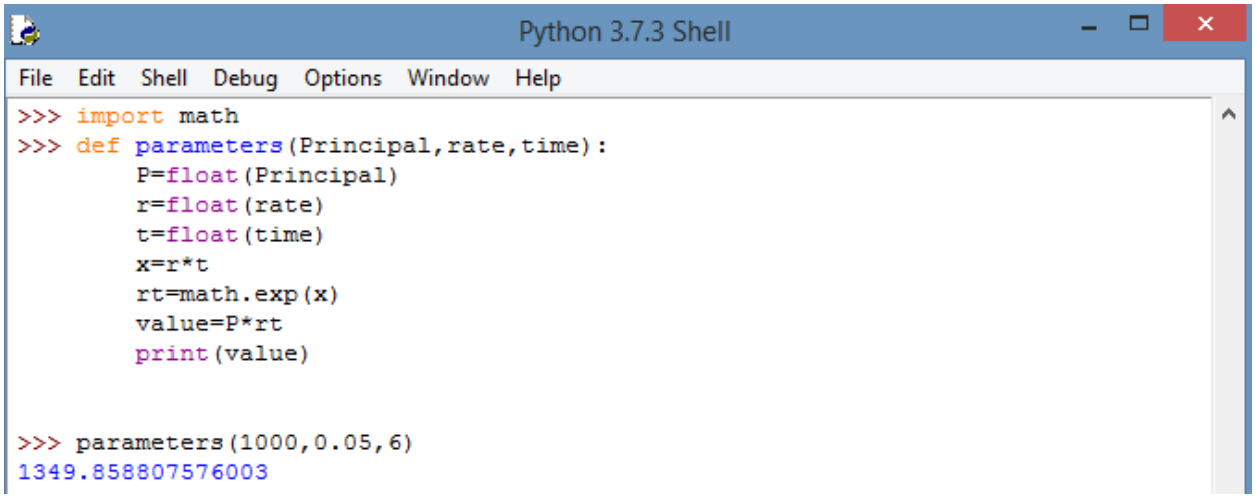
The following python program takes two integers m and d from the command line and writes *True* if day d of month m is between March 20 and June 20, and *False* otherwise as shown below:




To show the code executes as required, we use the following dates: May 5 and December 10 to test if they lie between March 20 and June 20 in each case as examples to the solution as shown above.

**6.**

The following python program calculates and writes the amount of money you would have if you invested it at a given interest rate (compounded continuously), taking the number of years t, the principal P, and the annual interest rate r as command-line arguments. The desired value is given by the formula $Pe^{rt}$.




To show the code executes as required, we use the following examples:

In the first example as shown above, we consider the Principal to be 1000 at an annual interest rate of 5% (or 0.05) for a period of 6 years to obtain the desired value: 1349.858807576003.

In the second example as shown below, we consider the Principal to be 500 at the same annual interest rate of 5% (or 0.05) for a period of 7.5 years to obtain the desired value: 727.4957073091007.