

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:

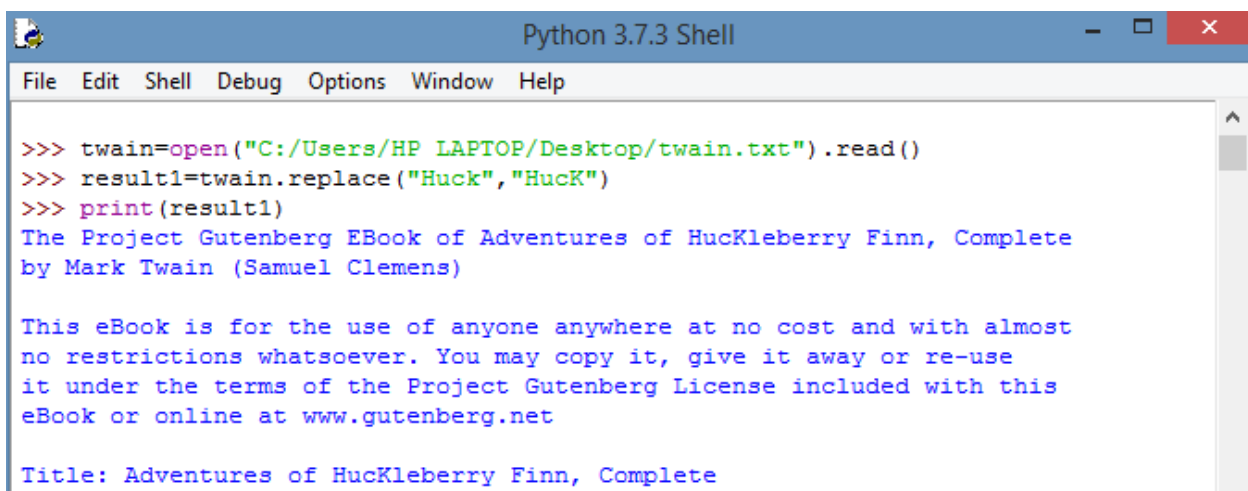
```
open("C:/Users/HP LAPTOP/Desktop/twain.txt").read()
```

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  
  
>>> twain=open("C:/Users/HP LAPTOP/Desktop/twain.txt").read()  
>>> result1=twain.replace("Huck", "HucK")  
>>> print(result1)  
The Project Gutenberg EBook of Adventures of HucKleberry Finn, Complete  
by Mark Twain (Samuel Clemens)  
  
This eBook is for the use of anyone anywhere at no cost and with almost  
no restrictions whatsoever. You may copy it, give it away or re-use  
it under the terms of the Project Gutenberg License included with this  
eBook or online at www.gutenberg.net  
  
Title: Adventures of HucKleberry Finn, Complete
```

```
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.

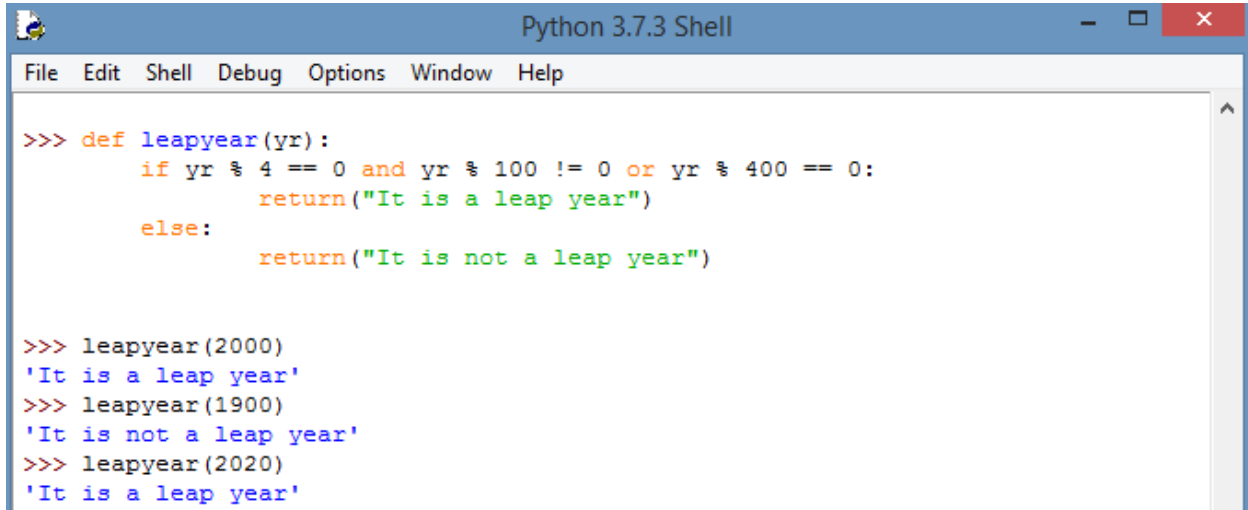
CHAPTER XIII. Escaping from the Wreck.--The Watchman.--Sinking.

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:

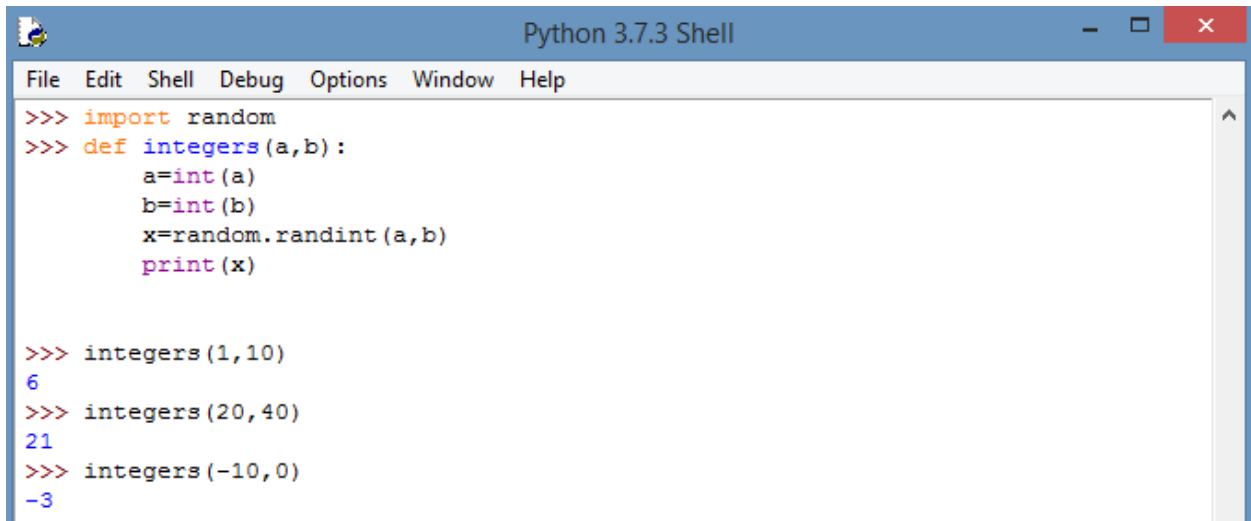
A screenshot of a Python 3.7.3 Shell window. The window has a blue title bar with the text "Python 3.7.3 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area of the window is a white text editor with a vertical scrollbar on the right. It contains the following Python code:

```
>>> def leapyear(yr):  
    if yr % 4 == 0 and yr % 100 != 0 or yr % 400 == 0:  
        return("It is a leap year")  
    else:  
        return("It is not a leap year")  
  
>>> leapyear(2000)  
'It is a leap year'  
>>> leapyear(1900)  
'It is not a leap year'  
>>> leapyear(2020)  
'It is a leap year'
```

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()*.



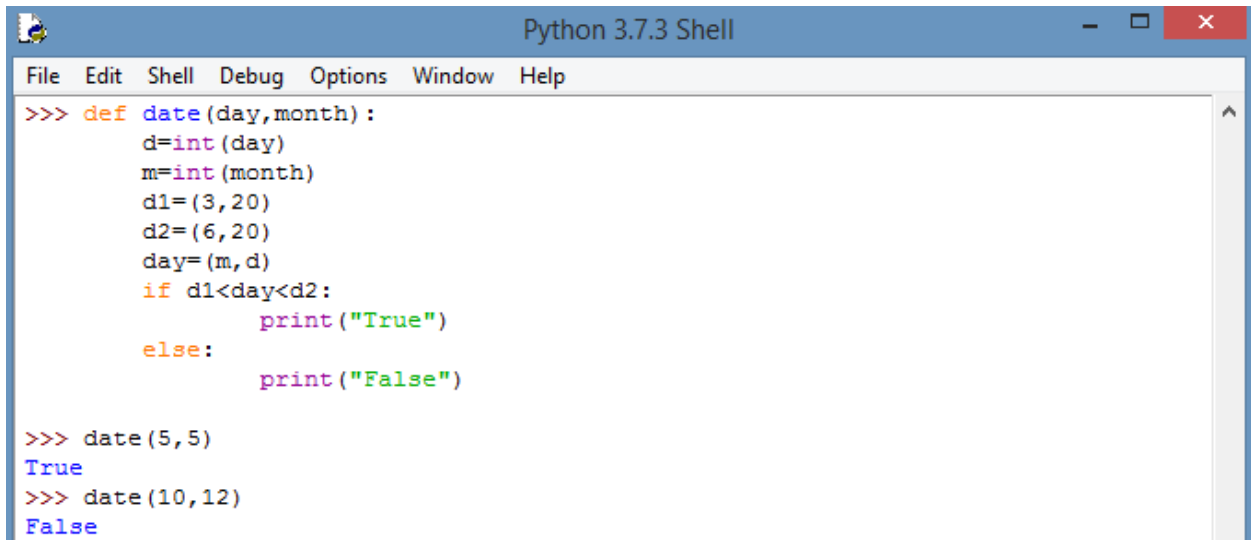
```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
>>> import random
>>> def integers(a,b):
    a=int(a)
    b=int(b)
    x=random.randint(a,b)
    print(x)

>>> integers(1,10)
6
>>> integers(20,40)
21
>>> integers(-10,0)
-3
```

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:

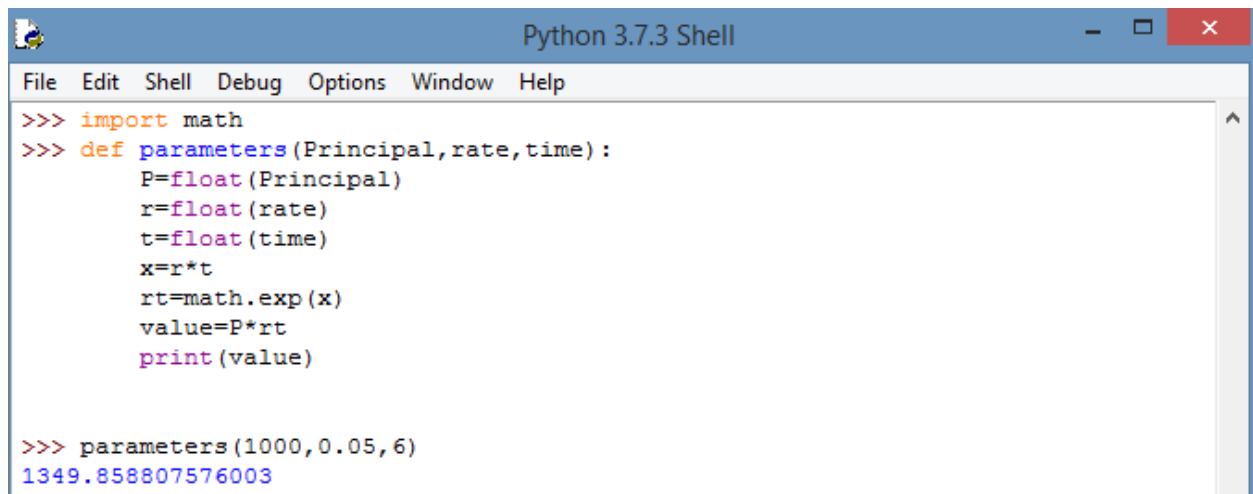
A screenshot of a Python 3.7.3 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main area contains the following Python code:

```
>>> def date(day,month):  
    d=int(day)  
    m=int(month)  
    d1=(3,20)  
    d2=(6,20)  
    day=(m,d)  
    if d1<day<d2:  
        print("True")  
    else:  
        print("False")  
  
>>> date(5,5)  
True  
>>> date(10,12)  
False
```

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} .

A screenshot of a Python 3.7.3 Shell window. The window has a blue title bar with the text "Python 3.7.3 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area contains the following Python code:

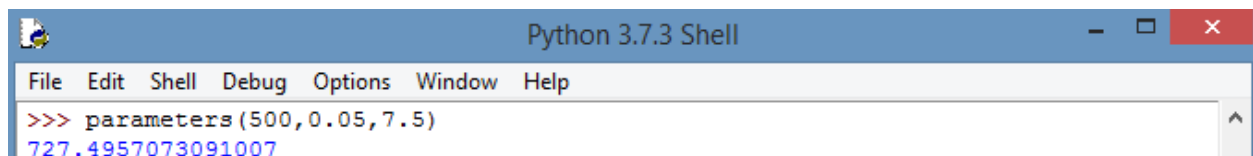
```
>>> import math
>>> def parameters(Principal,rate,time):
    P=float(Principal)
    r=float(rate)
    t=float(time)
    x=r*t
    rt=math.exp(x)
    value=P*rt
    print(value)

>>> parameters(1000,0.05,6)
1349.858807576003
```

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.

A screenshot of a Python 3.7.3 Shell window, similar to the one above. The main text area shows the execution of the parameters function with different arguments:

```
>>> parameters(500,0.05,7.5)
727.4957073091007
```