SI@UCF Introduction to Programming in Python Test #1 Solutions
6/15/2018

1) (5 pts) Write a **single statement** in python the produces the following output:

```
A\B/C
D/E\F
G\H/I
```

```
print("A\\B/C\nD/E\\F\nG\\H/I")
```

Grading: 1 pt print, 1 pt (), 1 pt "", 1 pt \n, 1 pt \\

2) (10 pts) What are the values of the following expressions in Python?

a) `23 - 3*5` **8**

b) `1623%100` **23**

c) `1 + 2 - (6*4//7)` **0**

d) `(4*(125/(3 + 7%12)))//8` **6**

e) `8 + 1234//1235 + 28000%14` **8**

Grading: 2 pts each all or nothing.

3) (12 pts) A cookie recipe that makes 24 cookies uses 2 eggs, 1 cup of butter and 6 ounces of brown sugar. (Note: The recipe uses other items, but we just care about these three.) Complete the program below so that it prompts the user to enter how many cookies she wants to make and prints out how many eggs, cups of butter and ounces of brown sugar are needed.

```
numCookies = int(input("How many cookies do you want to make?\n"))

numberOfEggs = (numcookies/24) * 2
cupsOfButter = (numcookies/24)
ouncesOfBrownSugar = (numcookies/24) * 6

print("You will need", numberOfEggs , "number of eggs.")
print("You will need", cupsOfButter , "cups of butter.")
print("You will need", ouncesOfBrownSugar , "ounces of brown sugar.")
```

Grading: 6 pts for the main fraction, 2 pts each for using the fraction to get to each of the items to print.

4) (12 pts) All leap years on Mars are divisible by 9. But, some years divisible by 9 are NOT leap years. Namely, if a year is divisible by 9, but also ends in a 0, it is NOT a leap year. Complete the program below so that it reads in a year on Mars and prints out if it is a leap year or not.

```
year = int(input("Enter a year (on Mars).\n"))

if year%9 != 0:
    print("Not a leap year.")
elif  year%9 == 0 and year%10 == 0:
    print("Not a leap year.")
else:
    print("It is a leap year.")
```

```
Grading: 4 pts for print no for all years not divisible by 9
         4 pts for printing yes for at least some yrs div by 9
         4 pts for getting the mod 90s correct.
```

5) (15 pts) Paula is considers two investment strategies every year. For one of the strategies, if she pays $1000 dollars up front, the rest of her money that remains doubles. For the other strategy, she pays $500 dollars up front and the rest of her money grows by 50%. (So, if she has $2400 at the beginning of the year, with strategy 1, she'll have ($2400-$1000)*2 = $2800. With strategy 2, she'll have ($2400-$500)*1.5 = $2850. Naturally, Paula is smart enough each year to choose the better of these two investment strategies. Complete the program below so that it reads in how much money Paula has at the beginning of the first year and how many years she wants to invest and calculates how much money she'll have at the end of her investment, assuming that each year she picks the better of these two strategies.

```
money = int(input("How many dollars are you investing?\n"))
numYears = int(input("How many years will you invest?\n"))


for i in range(numYears):
    if (money - 500)*1.5 > (money - 1000)*2:
        money =  (money - 500)*1.5
    else:
        money = (money - 1000)*2

print(money)
```

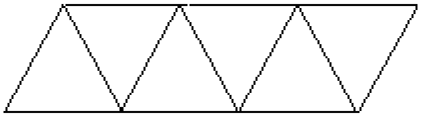
```
Grading: 3 pts loop through years, 4 pts each calculation,
         4 pts proper reassignment of money, 0 pts print
```

6) (10 pts) What is the output of the following Python program?

```
a = 8
b = 9
for i in range(5):
    c = b + 2*a
    print(a,b,sep=",",end=" : ")
    b = c//2
    a = a - c%a
```

8,9 : 7,12 : 2,13 : 1,8 : 1,5 : **Grading: 1 pt per value**

7) (20 pts) Write a python program below using the turtle so that it asks the user to enter a positive integer n which draws 2n equilateral triangles of side length 50 next to each other horizontally, one with the base on the bottom, the next on the top, and so forth. Here is the picture for n = 3:



```
import turtle

n = int(input("What is n?\n"))

for i in range(n):

    for j in range(3):
        turtle.forward(50)
        turtle.left(120)

    turtle.forward(50)

turtle.left(60)
turtle.forward(50)
turtle.left(120)
turtle.forward(50*n)
```

Grading: 6 pts for one triangle, 9 pts for repeating it so we get the bottom ones. 5 pts for finishing up the top ones.

8) (12 pts) Recall that the function random.randint(a,b) returns a random integer in between a and b, inclusive. Write a program that generates 500 random integers in between 1 and one million and then prints out the smallest and largest of all of the random integers generated.

```
import random

low = 1000000
high = 1

for i in range(500):

    tmp = random.randint(1, 1000000)

    if tmp > high:
        high = tmp
    if tmp < low:
        low = tmp

print(low, high)
```

```
Grading: 1 pt init values, 3 pts loop, 4 pts rand int,
         2 pt update min, 2 pt update max
```

9) (4 pts) What tasty breakfast item is Bagel King's specialty? **Bagels** **Grading: Give to all**