CS110A: Spring 2020 Name:
808 Kearny St. Chinatown Campus
(25 Pts) Solutions

1. (5 pts) Using Python for basic operations:

a. Construct a salutation sentence using the variable `userid` to create a greeting. For example, if `userid = "John Doe"` your computer might say: `"Welcome to Earth: John Doe"` (1 pt)

```
userid = "John Doe"
greeting = "Live Long and Propsper: "+userid
print(greeting)
```

b. Write code which draws a pentagon using the turtle library (2 pts)

```
import turtle

irma = turtle.Turtle()
pent_length = 100

for i in range(5):
 irma.forward(pent_length)
 irma.left(360/5)
```

d. Print out the square root of Pi with 4 digits of significant figures (1 pt)

*** As we discussed in class, a better word for this is mantissa or **precision** ***

```
testvalue = 3.14156**0.5
print('{:1.4f}'.format(testvalue))
```

e. Using a Truth Table, show whether the following expressions are equivalent (1 pt)

1) not (A and B) and not C

2) not C and not A or not B

We didn't go over Truth Tables in class, but if all you did was run these two boolean statements in repl.it and state the answer, you would have gotten the full point for this.

```
A = True
B = False
C = True

exprA = not (A and B) and not C
exprB = not C and not A or not B

print(exprA==exprB)
```

The correct answer is **Not Equivalent**

2. (15 pts) Invent some brand new dice:

   a. Print out all the **even** numbers between 0 and 100: (2 pts)

```
for i in range(50):
 print(2*i)
```

   Pay attention to the boundary conditions here at 0 and 50, so for example range(51) would be wrong.

   b. Print out all the **odd** numbers between 0 and 100: (2 pts)

```
for i in range(50):
 print(2*i+1)
```

   c. Using the random library, print out **10 random numbers** between 0 and 100: (2 pts)

   You could have used the built in randrange functions or a number of other ways to do this as well.

```
import random

for i in range(10):
 print(round(random.random()*100))
```

   d. Print out the **sum** of the first **10 even numbers** starting from 0 **using a for loop**: (2 pts)

```
accum = 0
for i in range(10):
 even_number = 2*i
 accum += even_number
print(accum)
```

e. Print out the **sum** of the first **10 even numbers** starting from 0 **without using a for loop**: (2 pts)

   Since you had the computer in front of you doing the exam, you could just look up the formula for a summation series. Or, you could try to rationalize it :)

   https://en.wikipedia.org/wiki/Arithmetic_progression

   10*(0+18)/2

   print(10*(0+18)/2)

f. In words, describe which solution is more efficient for generating the sum of 10 even numbers (part d. or part e.): (1 pt)

   You're just supposed to develop an intuition here but according to the classic methods of thinking like a computer scientist, I'd say that **part e is more efficient**. This is because it is more general, has less than 10 operations on just two values (the first and last). I'll read through solutions here though and look out for alternative explanations ;)

g. Carefully write down the nine numbers generated from the following piece of code: (1 pt)

```
 for i in range(1, 10):
     print (i*2%3+i*2%3+(i+1)//3)
```

4, 3, 1, 5, 4, 2, 6, 5, 3

h. Generate nine random numbers between 1 and 6 using the python random library: (2 pt)

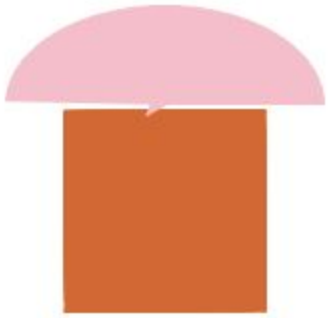
```
import random
for i in range(9):
      print int(random.random()*6+1)
```

i. In words: Describe which solution is more expensive (part g. Or part h.) for generating 10 numbers between 1 and 6. Would you use either of these methods for simulating a dice roll? (1 pt)

As far as I know, part h is more expensive. Although, refer to the Interactive Python notes about the nature of **pseudorandom** when using python. I would say that part g is cheating :) and the test question is not well written in that it should be “generating random numbers” so basically depending on your guys’ answers here I could take either part g or part h :)

Taking a break here before we get to the sprinkles question!

3. (5 pts) This cupcake needs some sprinkles. Do the best you can!



```
import turtle
import math

t = turtle.Turtle()
SIZE_OF_BASE = 100

# Chocolate base
t.color("chocolate", "chocolate")
t.begin_fill()
for i in range(4):
 t.forward(SIZE_OF_BASE)
 t.left(90)
t.end_fill()

# Frosting
t.left(90)
t.forward(SIZE_OF_BASE)
t.right(90)
# Ellipse: Convert Polar Coordinates
center_of_frosting_x = SIZE_OF_BASE/2
center_of_frosting_y = SIZE_OF_BASE+3

# Custom code to draw an ellipse
t.color("pink")
t.begin_fill()
t.goto(center_of_frosting_x,
center_of_frosting_y)
STEPSIZE = 2
RADIUSA = SIZE_OF_BASE*0.8
RADIUSB = SIZE_OF_BASE/2
for angle in range(0, 180, STEPSIZE):
 x =
RADIUSA*math.cos(math.radians(angle))+center_
of_frosting_x
 y =
RADIUSB*math.sin(math.radians(angle))+center_
of_frosting_y
 t.goto(x, y)
t.goto(center_of_frosting_x,
center_of_frosting_y)
t.end_fill()
```

#Sprinkles

The End

“Any sufficiently advanced technology is indistinguishable from magic” --Arthur C. Clarke

Clarke's First Law: When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong.

[[This Page Left Intentionally Blank]]