

Part 3

August 2017

1. Guess the results of the following statements (note: there are a couple of tricks, you may need to import something to avoid receiving an error message):

- a) `1==2`
- b) `1=2`
- c) `1<2`
- d) `pi == 3.141592`
- e) `1 = True`
- f) `1 == True`
- g) `0 == False`

```
from math import pi
print(1==2)
# print("1=2") #invalid
print(1<2)
print(pi==3.141592)
# print("1=True") #invalid
print(1==True)
# print(0=False) # invalid
```

```
## False
## True
## False
## True
```

2. Write a program that allows the user to enter a positive integer. If this number is even, print out its square. If it is odd, print out its cube. Hint: user input should be converted to an int; consider the `int()` method.

```
n = int(input("Enter a positive integer: "))
if n%2 == 0:
    print(n**2)
elif n%2 == 1:
    print(n**3)
```

3. Write a program that allows the user to enter two positive integers. If one number divides the other, print "yes", otherwise print "no".

```
n1 = int(input("Enter a positive integer: "))
n2 = int(input("Enter another positive integer: "))
if n1 % n2 == 0 or n2 % n1 == 0:
    print('yes')
else:
    print('no')
```

4. Similarly to activity 4 and 5, take an inputted number and make a program that will print out whether it is odd/even, bigger/less than 100, zero/non-zero, negative/positive.

```
num = int(raw_input('Please enter a number: '))
if num%2 == 1:
    print('Your number is odd.')
if num%2 == 0:
    print('Your number is even.')
```

```

if num > 100:
    print('Your number is greater than 100.')
if num < 100:
    print('Your number is less than 100.')
if num == 0:
    print('Your number is 0.')
if num != 0:
    print('Your number is not equal to 0.')
if num < 0:
    print('Your number is negative.')
if num > 0:
    print('Your number is positive.')

```

5. Create two of your very own infinite loops and explain why they'll be infinite.

```

count=0
while count<2:
    print(count) #doesn't increment the counter

badlist=[1,2,3]
for i in badlist:
    badlist.append(i) #increasing the length of the list each time

```

6. Create a conversion program that takes two user inputs: a number and a unit of mass ('kg', 'lb') in string form. Tell the user using the input function what they must enter and in what format. Make a program that takes a value and a unit, converts it to the other unit and then prints the result.

For example, say the user inputs 5 and 'lb'; your program would return '5 lb is xxx kg'

For some extra work instead of using only units of mass make the input any metric unit or any imperial unit and convert to the other system

For example, metres to yards, etc...

```

num = int(input('Please enter a number. In interger form.'))
unit = input('Please enter starting unit.')
if unit == 'lb':
    new_num = 0.45359237 * num
    print(num, unit, 'is', new_num, 'kg')
elif unit == 'kg':
    new_num = 2.20462262 * num
    print(num, unit, 'is', new_num, 'lb')
else:
    print('Incompatible units')
#for the extra work portion of the question simply add more if statements

```

7. Make a guessing game of your own similar to activity 7, but that guesses two numbers, not just one.

```

from random import randint
num_1=randint(0,100)
num_2=randint(0,100)

got_it = False
while got_it == False:
    guess1 = int(input('Enter an integer : '))
    guess2 = int(input('Enter a second integer : '))

```

```
if guess1 < num_1:
    print('A number is higher than that. Try again.')
elif guess2 < num_2:
    print('A number is higher than that. Try again.')
elif guess1 > num_1:
    print('A number is lower than that. Try again.')
elif guess2 > num_2:
    print('A number is lower than that. Try again.')

if num_1 == guess1 and num_2 == guess2:
    print('Congratulations, you guessed them.')
    print('(but you do not win any prizes!)')
    got_it = True
```

8. Create a program that counts from 0 to 25 and then back down to 0. Each time you add a number have the program print the number it is at. This program should be no more than 10 lines.

```
for i in range(26):
    print(i)
for i in range(26)[::-1]:
    print(i)
```

```
## 0
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 25
## 24
## 23
## 22
## 21
## 20
```

```
## 19
## 18
## 17
## 16
## 15
## 14
## 13
## 12
## 11
## 10
## 9
## 8
## 7
## 6
## 5
## 4
## 3
## 2
## 1
## 0
```

9. Write a function `sum` that takes two integer parameters `a` and `b` and returns the sum of squares of all the integers `n` such that $a \leq n \leq b$.

```
def sum(a, b):
    sum = 0
    for i in range(a,b+1):
        sum += i*i
    return sum

print(sum(1,5)) #55
```

```
## 55
```

10. Write a function `triangle` that takes three real parameters which represent the lengths of three sides of a triangle. The function should return `True` if the triangle is right-angled and `False` otherwise.

```
def triangle(a,b,c):
    #right angled triangles satisfy a2+b2=c2
    #for some combination of a,b,c
    right_angled = False
    if a**2 + b**2 == c**2 or\
        b**2 + c**2 == a**2 or\
        c**2 + a**2 == b**2:
        right_angled = True
    return right_angled

# tests
print("example output")
print("triangle(1,2,3) =",triangle(1,2,3)) #false
print("triangle(3,4,5) =",triangle(3,4,5)) #true
```

```
## example output
## triangle(1,2,3) = False
## triangle(3,4,5) = True
```

11. a. Write a function `factorial` that takes a nonnegative integer parameter and returns its factorial; we

define $0! = 1$.

```
def factorial(n):  
    fact = 1  
    for i in range(1,n+1):  
        fact *= i  
    return fact
```

- b. The number e can be defined as the limit of the infinite sum $1/0! + 1/1! + 1/2! + 1/3! + 1/4! + \dots$. Write a function `compute_e` which takes a positive integer parameter. This parameter indicates the number of terms that are to be summed (i.e. `compute_e(3)` should produce the sum $1/0! + 1/1! + 1/2!$). Sum the terms and return the approximate value of e . Call/use the function you wrote in part a in your solution.

```
def compute_e(n):  
    e_val = 0  
    for j in range(0,n):  
        e_val += 1./factorial(j)  
    return e_val
```