**1. Write and run a Python program that outputs the value of each of the following expressions:**

**5.0/9.0**

**5.0/9**

**5/9.0**

**5/9**

**9.0/5.0**

**9.0/5**

**9/5.0**

**9/5**

**now please try it with the truncated division operator:**

**5.0//9.0**

**5.0//9**

**5//9.0**

**5//9**

**9.0//5.0**

**9.0//5**

**9//5.0**

**9//5**

**Based on your results, what is the rule for arithmetic operators when integers and floating point numbers are used?**

>>> 5.0/9.0

0.5555555555555556

>>> 5.0/9

0.5555555555555556

>>> 5/9.0

0.5555555555555556

>>> 5/9

0.5555555555555556

>>> 9.0/5.0

1.8

>>> 9.0/5

1.8

>>> 9/5.0

1.8

>>> 9/5

1.8

>>> 5.0//9.0

0.0

>>> 5.0//9

0.0

>>> 5//9.0

0.0

>>> 5//9

0

>>> 9.0//5.0

1.0

>>> 9.0//5

1.0

>>> 9//5.0

1.0

>>> 9//5

1

**2. Write and run a Python program that asks the user for a temperature in Celsius and converts and outputs the temperature in Fahrenheit. (Use the formula given in the example above and solve for tempF in terms of tempC.)**

def main():

tempC = int(input("Enter the temperature in Celsius: "))

tempF = (tempC \* (9/5))+32

print(tempC, "C = ",tempF, " F")

main()

**OUTPUT:**

======================== RESTART: C:\Users\navya\Documents\python3.py =======================

Enter the temperature in Celsius: 40

40 C = 104.0 F

**3. Here is an algorithm to print out n! (n factorial) from 0! to 19!:**

**1. Set f = 1**

**2. Set n = 0**

**3. Repeat the following 20 times: # use range(1,20) although range(20) will also work but not match #5**

**a. Output n, "! = ", f**

**b. Add 1 to n**

**c. Multiply f by n**

**Using a for loop, write and run a Python program for this algorithm.**

f =1

n =0

for i in range(1,20):

n += 1

f \*= n

print(n,"!=", f)

**OUTPUT:**

===================== RESTART: C:\Users\navya\Documents\factorial sum.py ====================

1 != 1

2 != 2

3 != 6

4 != 24

5 != 120

6 != 720

7 != 5040

8 != 40320

9 != 362880

10 != 3628800

11 != 39916800

12 != 479001600

13 != 6227020800

14 != 87178291200

15 != 1307674368000

16 != 20922789888000

17 != 355687428096000

18 != 6402373705728000

19 != 121645100408832000

**4. Modify the program above using a while loop so it prints out all of the factorial values that are less than 1 billion. (You should be able to do this without looking at the output of the previous exercise.)**

f = 1

n = 0

while f< 1000000000:

print(n, "!=", f)

n+=1

f\*=n

**OUTPUT:**

=================== RESTART: C:\Users\navya\Documents\pythonfactorials.py ===================

0 != 1

1 != 1

2 != 2

3 != 6

4 != 24

5 != 120

6 != 720

7 != 5040

8 != 40320

9 != 362880

10 != 3628800

11 != 39916800

12 != 479001600

1. **Repeat #3 above using list comprehensions. Please see these increasingly more sophisticated references for list comprehensions: Simple, Advanced, Complex (has a hint for this problem).**

def solution():

f = 1

n =1

while True:

f \*= n

yield f

n += 1

factorial = solution()

result = [next(factorial) for i in range(1, 20)]

print(result)

**OUTPUT:**

==================== RESTART: C:\Users\navya\Documents\list factorial.py ====================

[1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, 39916800, 479001600, 6227020800, 87178291200, 1307674368000, 20922789888000, 355687428096000, 6402373705728000, 121645100408832000]

**5. Suppose you have a dictionary of dictionaries, or a 2D dictionary, called banditos as follows:**

**alonzo = {"age": 10, "height": 42, "weight": 175, "instrument":"fiddle" }**

**turing = {"age": 41, "height": 70, "weight": 160, "instrument": "theremin"}**

**bertha = {"age": 32, "height": 97, "weight": 587, "instrument": "cello"}**

**tinkerB = {"age":100, "height": 4, "weight": 0.5, "instrument": "cello"}**

**banditos = {"Alonzo": alonzo, "Turing": turing, "Bertha": bertha, "TinkerB": tinkerB}**

**We can access the sub-dictionary, or vector, of Alonzo's attributes as follows:**

**banditos["Alonzo"]**

**{"age": 10, "height": 42, "weight": 175, "instrument ": "fiddle" }**

**We can also return a specific feature of Alonzo's as:**

**banditos["Alonzo"]["age"]**

**10**

**For the above dictionary, banditos, write a function that, when given the dictionary, returns a list of users who play a certain instrument. E.g., here's a sample run and its result:**

**find\_players(banditos, "cello")**

**["Bertha", "TinkerB"]**

alonzo ={"age": 10, "height": 42, "weight": 175, "instrument":"fiddle"}

turing ={"age": 41, "height": 70, "weight": 160, "instrument": "theremin"}

bertha ={"age": 32, "height": 97, "weight": 587, "instrument": "cello"}

tinkerB ={"age":100, "height": 4, "weight": 0.5, "instrument": "cello"}

banditos = {"Alonzo" : alonzo, "Turing" : turing, "Bertha" : bertha, "TinkerB" : tinkerB}

def find\_players(banditos,instrument):

result = list(banditos.values())

length = len(result)

for i in range(length):

if result[i].get("instrument")==instrument:

print(list(banditos.keys())[i])

find\_players(banditos, "cello")

**OUTPUT:**

====================== RESTART: C:/Users/navya/Documents/5thquestion.py =====================

Bertha

TinkerB