**EXTRA PROGRAMS**

1. **Working with Data Structure - String**

**a)** Write a program to count occurrences of all characters within a string.

string = input("Enter the string: ")

d = {}

for s in string:

if s in d:

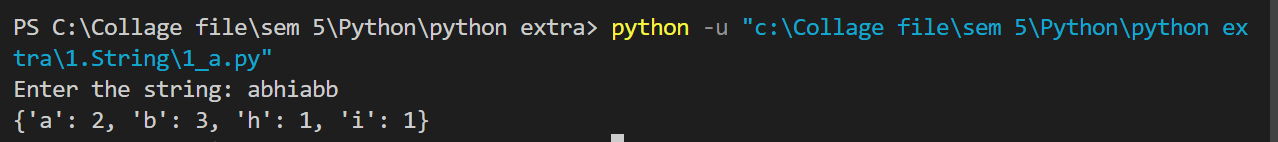
d[s] = d[s] + 1

else:

d[s] = 1

print(d)

**Output:-**

****

**b)** Write a program to reverse a given string.

string = input("Enter the string: ")

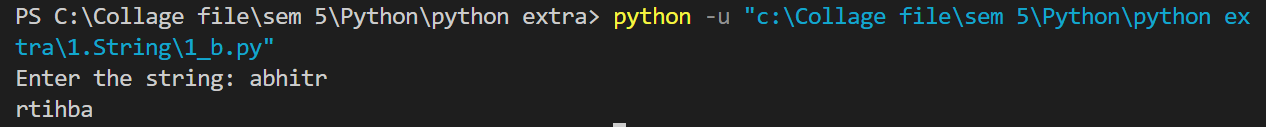
s = ''

for i in range(0, len(string)):

s = s + string[len(string) - i - 1]

print(s)

**Output:-**

****

**c)** Write a program to Find the last position of a given substring.

string = input("Enter the string: ")

find = input("Enter the substring: ")

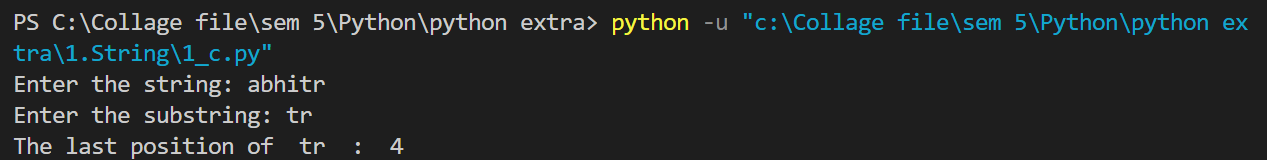
if find in string:

print("The last position of ",find, " : ",string.find(find))

else:

print(find, " does not exist.")

**Output:-**

****

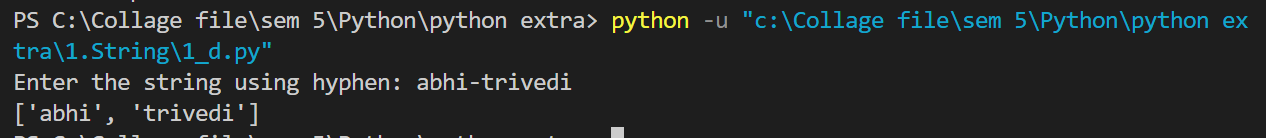
**d)** Write a program to Split a string on hyphens.

string = input("Enter the string using hyphen: ")

strArr = string.split("-")

print(strArr)

**Output:-**

****

**e)** Write a program to remove empty strings from a list of strings.

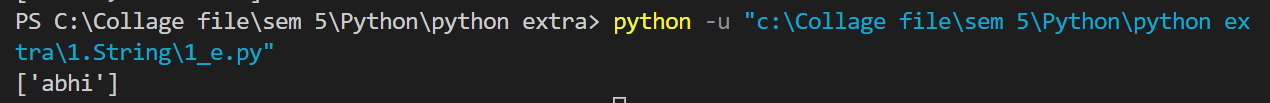
l = ["", "chaitya", ""]

while "" in l:

l.remove("")

print(l)

**Output:-**

****

**f)** Write a Program to calculate the sum and average of the digits present in a string.

string = input("Enter string: ")

sum, avg, count = 0, 0, 0

for s in string:

if s.isdigit():

sum = sum + int(s)

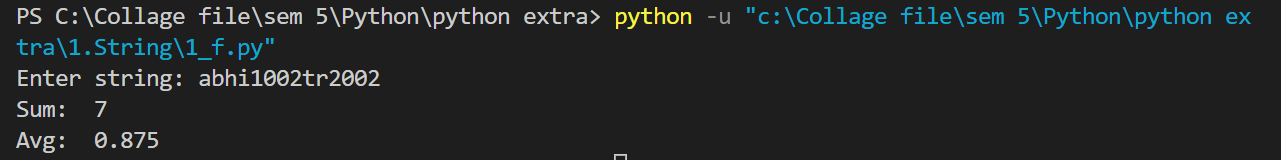
count = count + 1

avg = sum / count

print("Sum: ", sum)

print("Avg: ", avg)

**Output:-**

****

1. **Working with Data Structure – Tuple**

**a)**Write a program to create a tuple of all odd numbers from 10 to 20.

l = []

for i in range(10, 21):

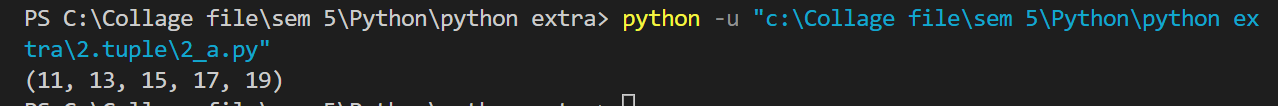
if i%2 != 0:

l.append(i)

t = tuple(l)

print(t)

**Output:-**

****

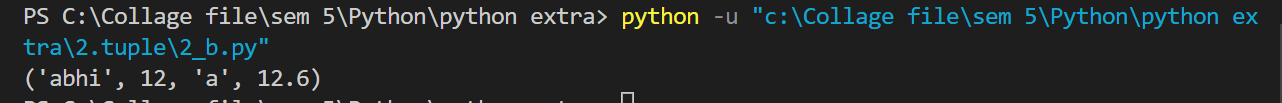
**b)** Write a program to create tuple with different data types and print them.

l = ["chaitya", 12, 'a', 12.6]

t = tuple(l)

print(t)

**Output:-**

****

**c)** Write a program to covert user given string into tuple with words as elements.

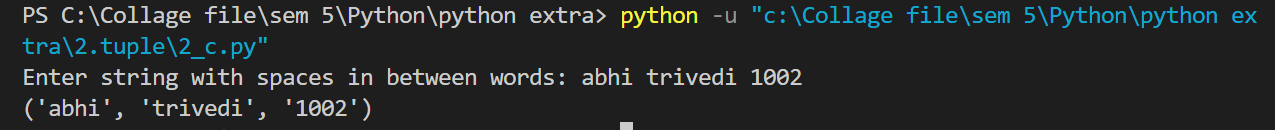
string = input("Enter string with spaces in between words: ")

l = string.split(" ")

t = tuple(l)

print(t)

**Output:-**

****

**d)** Write a program to find the repeated items of a tuple

t = ("chaitya", "yash", 12, 12)

d = {}

for i in t:

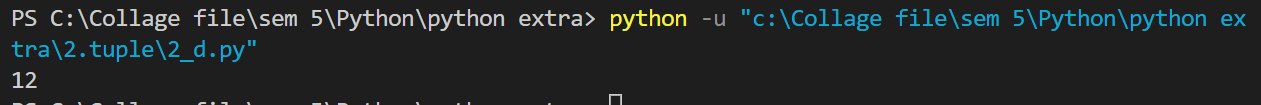
if i in d:

print(i)

else:

d[i] = 0

**Output:-**

****

**e)** Write a program to check whether an element exists within a tuple.

t = ("chaitya", "yash", "12", "12")

find = input("Enter the item: ")

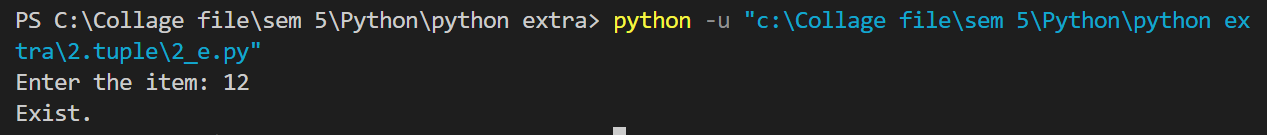
if find in t:

print("Exist.")

else:

print(find, " is not in the tuple.")

**Output:-**

****

**f)** Write a Python program to convert a list of tuples into a dictionary.

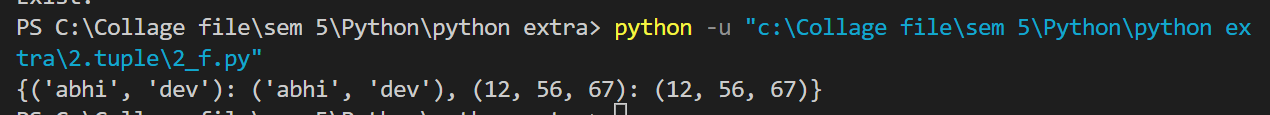
l1 = [("abhi", "dev"), (12, 56, 67)]

l2 = [("abhi", "dev"), (12, 56, 67)]

d = dict(zip(l1, l2))

print(d)

**Output:-**

****

**g)** Write a Python program to find the index of an item of a tuple

t = ("chaitya", "yash", "12", "12")

find = input("Enter the item: ")

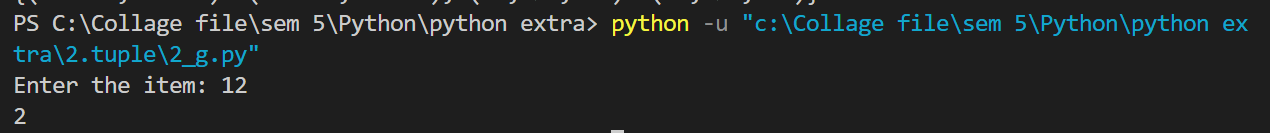
if find in t:

print(t.index(find))

else:

print(find, " is not in the tuple.")

**Output:-**

****

1. **Anonymous Functions – Lambda, Map, Filter, Reduce**

**a)**Write a Python program to find maximum of two user given numbers using lambda function.

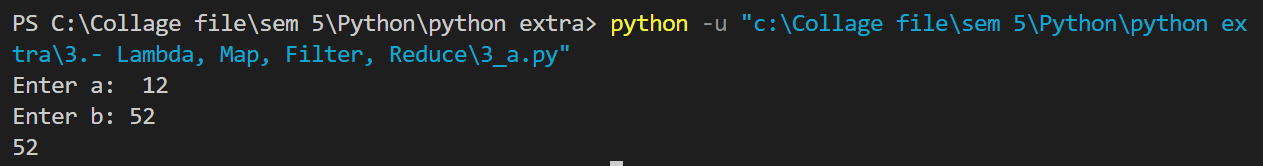
max = lambda a,b : a if a>b else b

a = int(input("Enter a: "))

b = int(input("Enter b: "))

print(max(a, b))

**Output:-**

****

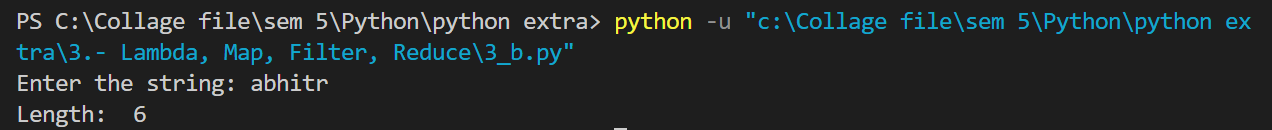
**b)** Write a Python program to find length of user given string using lambda function.

string = input("Enter the string: ")

length = lambda s : len(s)

print("Length: ", length(string))

**Output:-**

****

**c)** Consider a list of 1 to n (where n is user given number) and use map function to generate factorial of each numbers of list.

def factorial(n):

if n==1:

return 1

else:

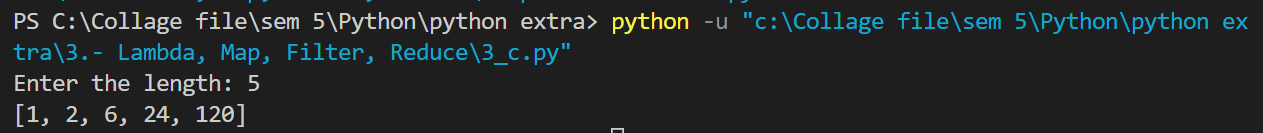
return n\*factorial(n-1)

n = int(input("Enter the length: "))

fact = list(map(factorial, range(1, n+1)))

print(fact)

**Output:-**

****

**d)** Consider a list consisting the marks of Physics subject of 10 students. Use reduce function to calculate the average of marks from the list.

from functools import reduce

import operator

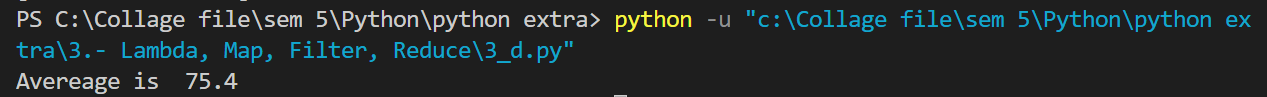
physics = [90, 99, 34, 56, 52, 78, 89, 78, 78, 100]

def average(l):

return reduce(lambda x,y : x+y , l) / len(l)

print("Avereage is ", average(physics))

**Output:-**

****

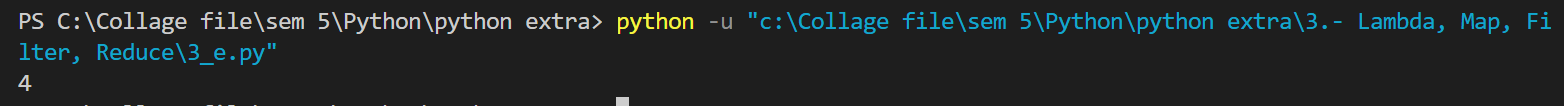
**e)** Write a Python Program to Find the Number Occurring Odd Number of Times using Lambda expression and reduce function.

from functools import reduce

l = [4, 4, 4, 8, 1, 1, 8]

ans = reduce(lambda x, y : x^y, l)

**Output:-**

****

**f)** Write a Python program to add two lists consisting of first names and last names using map function .

Ex.

Input : L1 = [“Patel” , “Desai”, “Shah”] L2 = [“Mansi”, “Pooja”,”Rahul”]

Output : [“PatelMansi” , “DesaiPooja” , ”ShahRahul”]

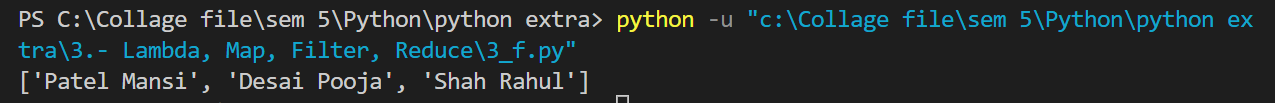
L1 = ["Patel" , "Desai", "Shah"]

L2 = ["Mansi", "Pooja","Rahul"]

ans = list(map(lambda x, y: x+" "+y, L1, L2))

print(ans)

**Output:-**

****

1. **Modules**

**a)** Write a Python Program to find the day of the week of a given date

from datetime import date.

def day(s):

days = {

0:"Monday",

1:"Tuesday",

2:"Wednesday",

3:"thursday",

4:"Friday",

5:"Satureday",

6:"Sunday",

}

return str(days[s])

d = int(input("Enter the date: "))

month = int(input("Enter the month: "))

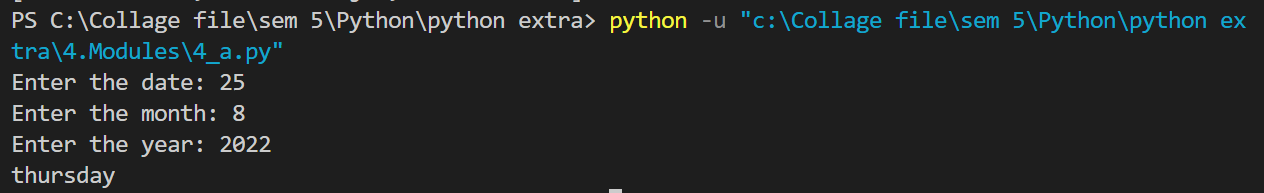
year = int(input("Enter the year: "))

myDate = date(year, month, d)

s = myDate.weekday()

print(day(myDate.weekday()))

**Output:-**

****

**b)**Write a Python Program to calculate number of days between two given dates.

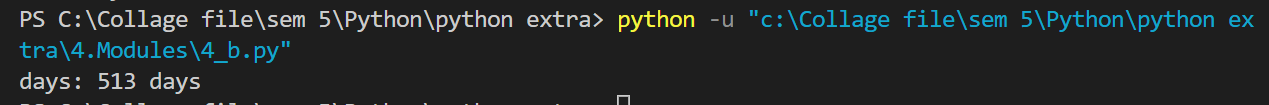
from datetime import date

date\_1 = date(2023, 9, 10)

date\_2 = date(2025, 2, 4)

print("days:", (date\_2 - date\_1).days, "days")

**Output:-**

****

**c)** Write a program to display Calendar of given month.

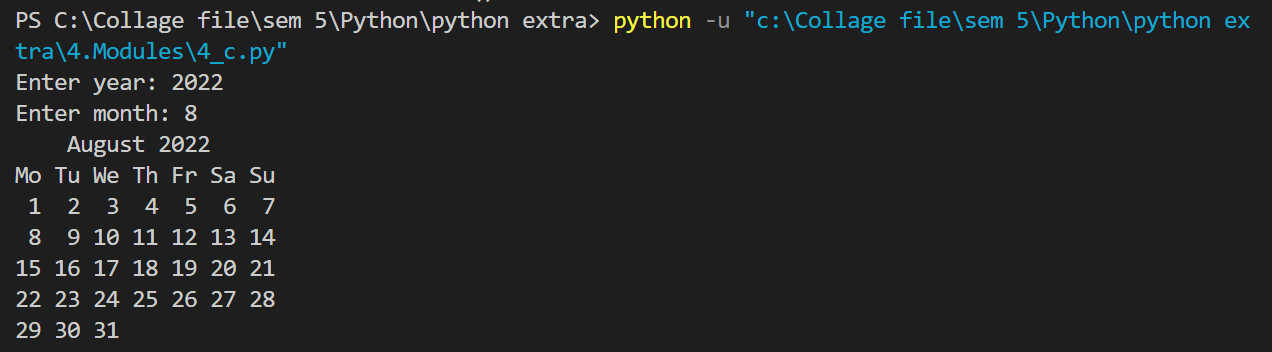
import calendar as cal

year = int(input("Enter year: "))

month = int(input("Enter month: "))

print(cal.month(year, month))

**Output:-**

****

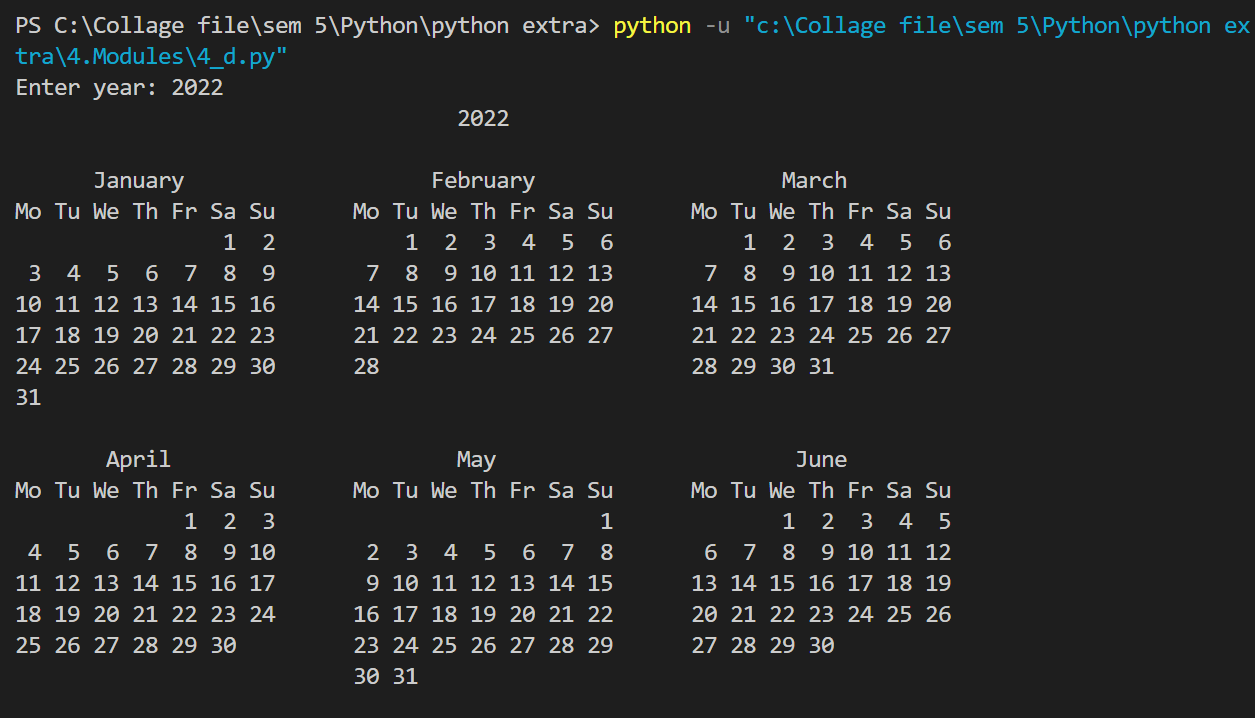
**d)** Write a program to display Calendar of given year.

import calendar as cal

year = int(input("Enter year: "))

print(cal.calendar(year))

**Output:-**

****