

**MCA SEC-C**

**NAME :- RITIK KAUSHIK**

**ROLL NO :- 48**

**UNIVERSITY ROLL NO :- 2284200171**

**SUBJECT :- Python**

**SEMESTER :- First Semester**

**SUBJECT TEACHER : - Dr. Ankur Sir**

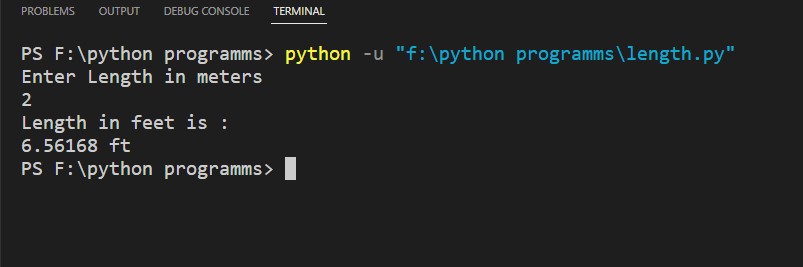
**Q1-WAP to enter the length in Meter and then convert into Feet and finally print**

n=float(input("Enter Length in meters\n"))

print("Length in feet is :")

print(f"{n\*3.28084} ft")

**OUTPUT**



|  |
| --- |
| **Q2-WAP to input two numbers and then swap these two numbers using** |
| **(i) 3 variables** |
| **(ii) 2 variables**  Using 2 variables  n1=int(input("Enter Two Numbers\n"))  n2=int(input())  n3=n1  n1=n2  n2=n3  print("After Swapped")  print(f"n1 = { n1 } , n2 = { n2 }")    swapped.jpg  Using 3 Variables  n1=int(input("Enter Two Numbers\n"))  n2=int(input())  n1,n2=n2,n1  print("After Swapped")  print(f"n1 = { n1 } , n2 = { n2 }")  swapped.jpg  **Q3-WAP to enter the time in Seconds and**  **then convert it into Hours,Minutes**  **and Seconds**  def convert\_to\_preferred\_format(sec):  sec = sec % (24 \* 3600)  hour = sec // 3600  sec %= 3600  min = sec // 60  sec %= 60  print("The value in hours:",hour)  print("The value in minutes:",min)  print("The value in second:",sec)  return "%02d:%02d:%02d" % (hour, min, sec)  n = 4502  print("Time in preferred format :-",convert\_to\_preferred\_format(n))  **OUTPUT**  time.jpg   |  | | --- | | **Q4-WAP to enter the amount in Rs. and suppose there are currency**  **notes of 500,100,50,20,10,2,1.Find the number of notes of each**  **denomination so that the counting of notes is minimum**  notes = (500,100,50,20,10,5,2,1)  amount = int(input("Enter Amount to be paid : "))  for C in notes:  count = amount//C  print("Note Value : ", C,'\tnumber of notes ',count)  amount = amount%C  **OUTPUT**  **NoteCount.jpg**  **Q5-WAP to find the GCD or HCF of two numbers.**  **Numbers will be asked from user**  num1 = int(input("Enter first number: "))  num2 = int(input("Enter second number: "))  i = 1  while(i <= num1 and i <= num2):  if(num1 % i == 0 and num2 % i == 0):  gcd = i  i = i + 1  print("The H.C.F. of", num1,"and", num2,"is", gcd, "".format(num1, num2, gcd))  **OUTPUT**  **HCF.jpg**  **Q6-WAP to find the LCM of two numbers. Numbers**  **Will be asked from user**  def compute\_lcm(x, y):  if x > y:  greater = x  else:  greater = y  while(True):  if((greater % x == 0) and (greater % y == 0)):  lcm = greater  break  greater += 1  return lcm  num1 = int(input("Enter first number: "))  num2 = int(input("Enter second number: "))  print("The L.C.M. is", compute\_lcm(num1, num2))  **OUTPUT**  **Lcm.jpg**  **Q7-** **WAP to find the factorial of a number where**  **number is asked from user**  n = int(input("Enter a number: "))  i = 1  f = 1  for i in range(i,n+1):  f = f\*i  print(f"Factorial is {f} ")  **OUTPUT**  **factorial.jpg**  **Q8-** **WAP to enter a number and find the reverse of it**  num = int(input("Enter a number\n"))  reversed\_num = 0  while num != 0:  digit = num % 10  reversed\_num = reversed\_num \* 10 + digit  num //= 10    print("Reversed Number: " + str(reversed\_num))  **OUTPUT**  **reverse.jpg** | |  | |  | |
|  |

**Q9-** **WAP to enter a number and then find the sum of digits**

def getSum(n):

sum = 0

for digit in str(n):

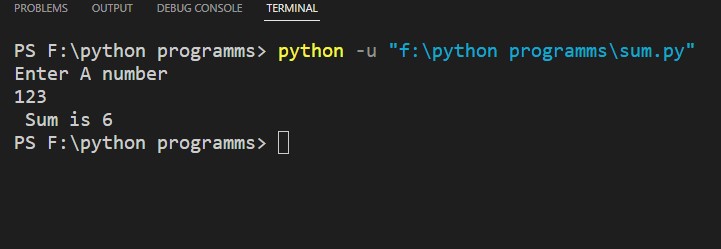
sum += int(digit)

return sum

n=int(input("Enter A number\n"))

print(f" Sum is {getSum(n)}")

**OUTPUT**

****

**Q10-** **Write a Python program to check whether a list contains a sublist.**

test\_list = [5, 6, 3, 8, 2, 1, 7, 1]

print("The original list : " + str(test\_list))

sublist = [8, 2, 1]

res = False

for idx in range(len(test\_list) - len(sublist) + 1):

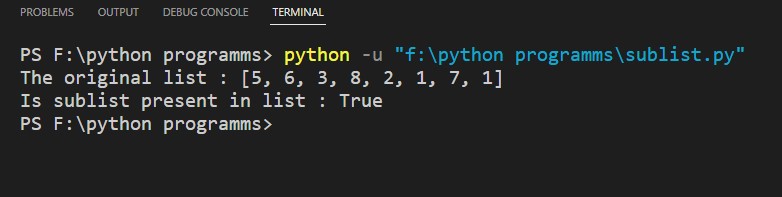
if test\_list[idx: idx + len(sublist)] == sublist:

res = True

break

print("Is sublist present in list : " + str(res))

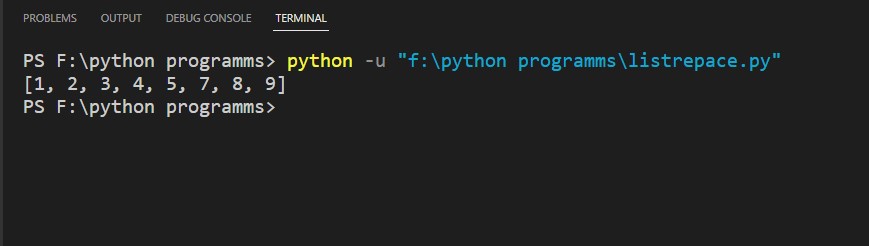
**OUTPUT**

****

**Q11- Write a Python program to replace the last element in a list with another list.**

|  |
| --- |
|  |
| Sample data : [1, 3, 5, 7, 9, 10], [2, 4, 6, 8] |
| Expected Output: [1, 3, 5, 7, 9, 2, 4, 6, 8]  l = [1,2,3,4,5,6]  l2 = [7,8,9]  l.remove(6)  for i in l2:  l.append(i)  print(l) |

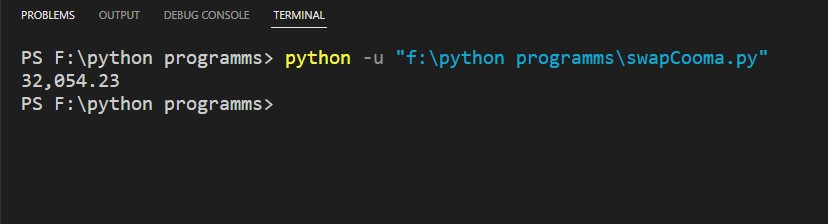
**OUTPUT**

****

**Q13-** **Write a Python program to swap comma and dot in a string.**

|  |
| --- |
|  |
| Sample string: "32.054,23" |
| Expected Output: "32,054.23"  str = "32.054,23"  m = str.maketrans  str = str.translate(m('.,',',.'))  print(str) |

**OUTPUT**

****

**Q14-** **Write a Program to print the inverted Dictionary.**

|  |
| --- |
| If D1={1:”Hello”, 2:”Bye”, 3:”World”}, |
| then print D2 as D2={“Hello”:1, ”Bye”:2, ”World”:3}  D1 = {1:"hello",2:"bye",3:"world"}  print(D1)  D1 = dict(map(reversed,D1.items()))  print(D1)  **OUTPUT**  **dictnorayinverted.jpg** |

|  |
| --- |
| **Q15-Write the code to do the following using numpy Package :** |
| **(i)Create a 2D array of dimension 3x4 and the elements will be random**  **numbers from 0 to1.** |
| **(ii)Create a 2D array of dimension and the elements will be all zero.** |
| **(iii) How to change the dimension from 3X4 to 4X3.** |

**Q16-** **Write a program to define a function** longest\_line(filename) **which accepts the filename and return the longest line in a file.**

max\_length = 0

max\_len\_line = ''

file = open("his.txt")

for line in file:

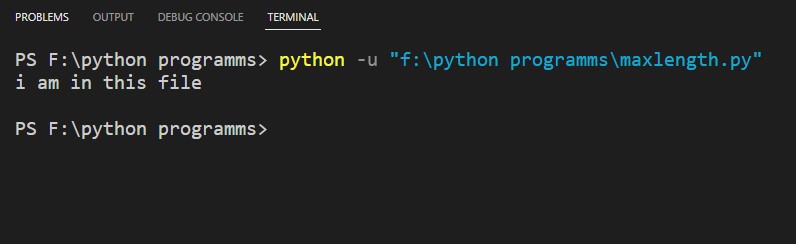
if(len(line) > max\_length):

max\_length = len(line)

max\_len\_line = line

print(max\_len\_line)

**OUTPUT**

****

**Q17-** **Write a Program which explains ZeroDivisonError.**

n1=int(input("Enter Two Numbers\n"))

n2=int(input())

try:

n3=n1/n2

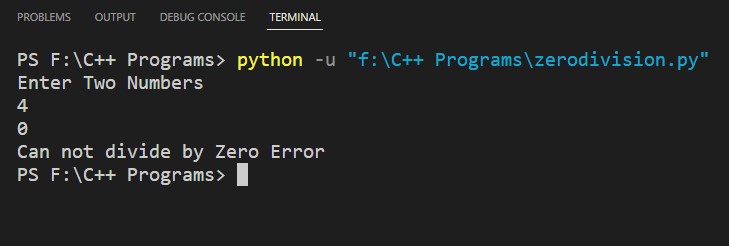
except ZeroDivisionError:

print("Can not divide by Zero Error")

else:

print(f"Division value is : {n3}")

**OUTPUT**

****

**Q18-** **Write a program to count the number of times “Hello” is occurring in the file.**

fileName='hello.txt'

with open(fileName) as f1:

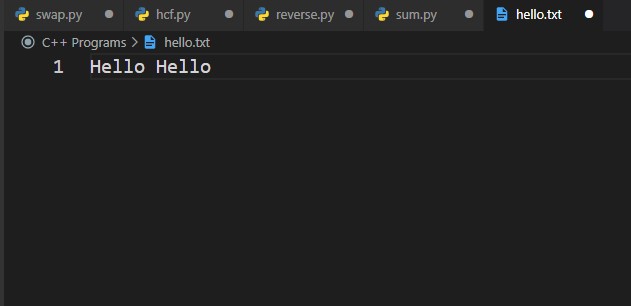
n=f1.read()

s=n.split()

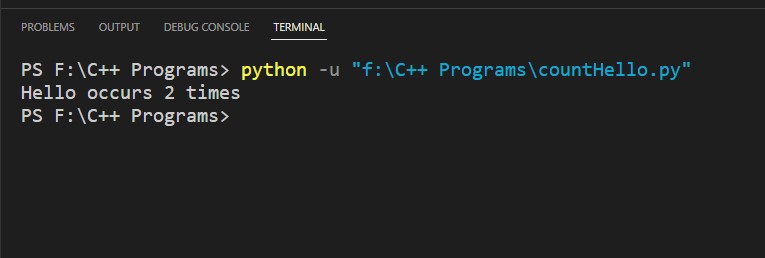
c=s.count("Hello")

print(f"Hello occurs {c} times")

**FILE**

****

**OUTPUT**

****

**Q19-** **Write a Python program to read a file (“input.txt”) of oil prices in England, Australia and India. Output the average oil price of each country to an output file output.txt. (Note: All fields are separated with one space)**

input='input.txt'

try:

with open(input) as f1:

data= f1.read()

except FileNotFoundError:

print("File Not Found")

output='output.txt'

try:

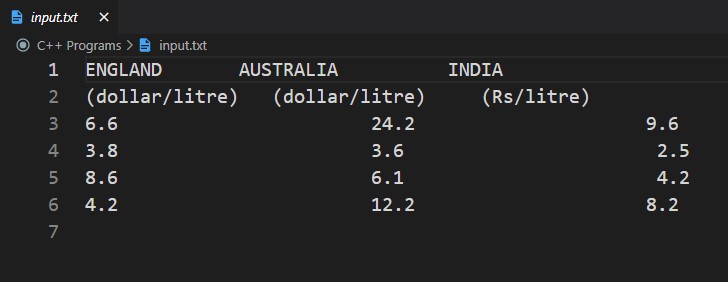
with open(output,'w') as f2:

f2.write(data)

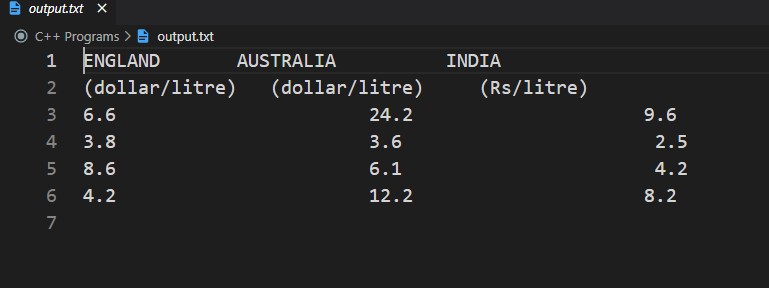
except FileNotFoundError:

print("File Not Found")

**FILE=input.txt**

****

**FILE=output.txt**

****

**Q20-** **WAP which makes the use of \_\_init\_\_() method and self keyword**

class Car:

def \_\_init\_\_(self,name,year,model):

self.name=name

self.year=year

self.model=model

def getData(self):

print(f"Name = {self.name}")

print(f"Year = {self.year}")

print(f"Model = {self.model}")

c1=Car("Hyundai",2022,"i20")

c1.getData()

**OUTPUT**

