**Experiment-1:**

**1.1) Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram**

**Program:**

weight=float(input())

pound=weight\*2.2

print(pound)

**Sample Input and Sample Output:**

2.5

5.5

**1.2) Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average**

**Program:**

a=float(input())

b=float(input())

c=float(input())

total=a+b+c

average=total/3

print(total,average)

**Sample Input and Sample Output:**

1

2

3.5

6.5 2.1666666666666665

**Experiment-2:**

**2.1) Write a program that uses a for loop to print the numbers 8,11,14,17,20, ....,83,86,89**

**Program:**

for i in range(8,90,3):

print(i,end=" ")

**Sample Input and Sample Output:**

8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77 80 83 86 89

**2.2) Write a program that asks the user for their name and how many times to print it. The program should print out the user’s name the specified number of times**

**Program:**

name=input()

n=int(input())

for \_ in range(n):

print(name)

**Sample Input and Sample Output:**

Rajesh

4

Rajesh

Rajesh

Rajesh

Rajesh

**Experiment-3:**

**3.1) Use a for loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.**

**Program:**

for i in range(1,int(input())+1):

for j in range(i):

print("\*",end=" ")

print()

**Sample Input:**

4

**Sample Output:**

\*

\* \*

\* \* \*

\* \* \* \*

**3.2) Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not**

**Program:**

import random

n=random.randint(1,10)

inp=int(input())

if inp==n:

print("You guessed the right number!")

else:

print("Too bad, you're wrong!")

**Sample Input and Sample Output:**

5

You guessed the right number!

**3.3) Write a program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and Not close otherwise**

**Program:**

n1=float(input())

n2=float(input())

if(abs(n1-n2)<0.001):

print("Close")

else:

print("Not close")

**Sample Input and Sample Output:**

5.001

5.0001

Close

**Experiment-4:**

**4.1) Write a program that asks the user to enter a word and prints out whether that word contains any vowels**

**Program:**

s=input()

for i in s:

if i in "aeiouAEIOU":

print("String contains vowels")

break

else:

print("String doesn't contain any vowels")

**Sample Input and Sample Output:**

Rajesh

String contains vowels

**4.2) Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe**

**Program:**

s1=input()

s2=input()

if len(s1)==len(s2):

s=""

for i in range(len(s1)):

s+=s2[i]+s1[i]

print(s)

else:

print("Strings are of not same length")

**Sample Input and Sample Output:**

abcde

ABCDE

AaBbCcDdEe

**4.3)Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be 1,000,000**

**Program:**

n=int(input())

print("The number seperated with commas is:{:,}".format(n))

**Sample Input and Sample Output:**

10000000

The number seperated with commas is:10,000,000

**4.4) In algebraic expressions, the symbol for multiplication is often left out, as in 3x+4y or 3(x+5). Computers prefer those expressions to include the multiplication symbol, like 3\*x+4\*y or 3\*(x+5). Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate**

**Program:**

s=list(input())

res=""

i=0

while(i<len(s)):

if(s[i]=='('):

index=s.index(')')

s2="".join(s[i:index+1])

res+="\*"+s2

i+=len(s2)

elif s[i].isalpha():

res+='\*'+s[i]

i+=1

else:

res+=s[i]

i+=1

print(res)

**Sample Input and Sample Output:**

3x+4y

3\*x+4\*y

**Experiment-5:**

**5.1) Write a program that generates a list of 20 random numbers between 1 and 100.**

**(a) Print the list.**

**(b) Print the average of the elements in the list.**

**(c) Print the largest and smallest values in the list.**

**(d) Print the second largest and second smallest entries in the list**

**(e) Print how many even numbers are in the list**

**Program:**

import random

l=random.sample(range(1, 100), 20)

print(\*l)

print(sum(l)/20)

l.sort()

print(max(l),min(l))

print(l[-2],l[1])

print(len([i for i in l if i%2==0]))

**Sample Input and Sample Output:**

34 14 29 71 52 51 92 8 46 44 22 86 28 36 31 72 20 23 47 81

44.35

92 8

86 14

13

**5.2) Write a program that asks the user for an integer and creates a list that consists of the factors of that integer**

**Program:**

n=int(input())

l=[]

for i in range(1,n+1):

if n%i == 0:

l.append(i)

print(\*l)

**Sample Input and Sample Output:**

12

1 2 3 4 6 12

**5.3) Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4**

**Program:**

import random

l=[]

for i in range(100):

n=random.randint(0,1)

l.append(n)

maxx=0

c=0

print(\*l)

for i in range(100):

if l[i]==0:

c+=1

maxx=max(c,maxx)

else:

c=0

print(maxx)

**Sample Input and Sample Output:**

1 0 1 0 1 0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1

5

**Experiment-6:**

**6.1) Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0]**

**Program:**

l=list(map(int,input().split()))

arr=[]

for i in l:

if i not in arr:

arr.append(i)

print(arr)

**Sample Input and Sample Output:**

1 1 2 3 4 3 0 0

[1, 2, 3, 4, 0]

**6.2) Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements,it is much shorter with lists and it is also easier to add new conversions if you use lists**

**Program:**

n=float(input())

k=int(input())

num=0

if k==1:

num=n\*12

elif k==2:

num=n\*0.33

elif k==3:

num=n\*0.000189394

elif k==4:

num=n\*304.8

elif k==5:

num=n\*30.48

elif k==6:

num=n\*0.3048

elif k==7:

num=n\*0.0003048

else:

num=-1

if(num==-1):

print("No conversion happened!")

else:

print(round(num,2))

**Sample Input and Sample Output:**

5.89

1

70.68

**6.3) Write a python script to perform following operations:**

**i) Create a matrix and print it**

**ii) Perform Addition of 2 matrices**

**iii) Perform multiplication of 2 matrices**

**Program:**

r1,c1=map(int,input().split())

a=[]

for i in range(r1):

l=list(map(int,input().split()))

a.append(l)

r2,c2=map(int,input().split())

b=[]

for i in range(r2):

l=list(map(int,input().split()))

b.append(l)

for i in a:

print(i)

for i in b:

print(i)

res=[[a[i][j]+b[i][j] for j in range(c1)] for i in range(r1)]

print("Addition of two matrices is:")

for i in res:

print(i)

print("Multiplication of two matrices is:")

res=[[0 for j in range(c1)] for i in range(r1)]

for i in range(r1):

for j in range(c2):

for k in range(r2):

res[i][j]+=a[i][k]\*b[k][j]

for i in res:

print(i)

**Sample Input and Sample Output:**

3 3

1 2 3

4 5 6

7 8 9

3 3

1 2 3

4 5 6

7 8 9

[1, 2, 3]

[4, 5, 6]

[7, 8, 9]

[1, 2, 3]

[4, 5, 6]

[7, 8, 9]

Addition of two matrices is:

[2, 4, 6]

[8, 10, 12]

[14, 16, 18]

Multiplication of two matrices is:

[30, 36, 42]

[66, 81, 96]

[102, 126, 150]

**Experiment-7:**

**7.1) Write a function called sum\_digits that is given an integer num and returns the sum of the digits of num.**

**Program:**

def sum\_digits(n):

s=0

while(n):

s+=(n%10)

n//=10

return s

print(sum\_digits(int(input())))

**Sample Input and Sample Output:**

1234

10

**7.2) Write a function called first\_diff that is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return -1**

**Program:**

def first\_diff(s1,s2):

for i in range(min(len(s1),len(s2))):

if(s1[i]!=s2[i]):

return i

return -1

print(first\_diff(input(),input()))

**Sample Input and Sample Output:**

Harsha

Harsha

-1

**7.3) Write a function called number\_of\_factors that takes an integer and returns how many factors the number has**

**Program:**

def number\_of\_factors(n):

s=0

for i in range(1,n+1):

if n%i == 0:

s+=1

return s

print(number\_of\_factors(int(input())))

**Sample Input and Sample Output:**

10

4

**7.4) Write a function called is\_sorted that is given a list and returns True if the list is sorted and False otherwise**

**Program:**

def is\_sorted(l):

li=sorted(l)

if(l==li):

return "True"

return "False"

print(is\_sorted(list(map(int,input().split()))))

**Sample Input and Sample Output:**

1 2 4 3 6

False

**Experiment-8:**

**8.1) Write a function called root that is given a number x and an integer n and returns x^(1/n) . In the function definition, set the default value of n to 2**

**Program:**

import math

def root(x,n=2):

return math.pow(x,1/n)

print(root(float(input())))

**Sample Input and Sample Output:**

25

5.0

**8.2) Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100**

**Program:**

def primes(n):

l=[]

i=2

while(n):

for j in range(2,int(i\*\*0.5)+1):

if i%j==0:

break

else:

l.append(i)

n-=1

i+=1

return l

print(\*primes(100))

**Sample Input and Sample Output:**

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541

**8.3) Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list.**

**(a) Do this using the sort method.**

**(b) Do this without using the sort method**

**Program:**

def mergesort(l1,l2):

i,j=0,0

l=[]

while(i<len(l1) and j<len(l2)):

if(l1[i]<l2[j]):

l.append(l1[i])

i+=1

else:

l.append(l2[j])

j+=1

while(i<len(l1)):

l.append(l1[i])

i+=1

while(j<len(l2)):

l.append(l2[j])

j+=1

return l

def merge(l1,l2):

return sorted(l1+l2)

l1=list(map(int,input().split()))

l2=list(map(int,input().split()))

print(\*merge(l1,l2))

print(\*mergesort(l1,l2))

**Sample Input and Sample Output:**

1 3 5 7

2 4 6 8

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8

**8.4) Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can’t exceed the number of occurrences of the letter in the user’s word.**

**Program:**

from itertools import permutations

s=input()

for i in range(2,len(s)):

for p in permutations(s,i):

print("".join(p),end=" ")

**Sample Input and Sample Output:**

car

ca cr ac arrcra

**Experiment-9:**

**9.1) Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons**

**Program:**

file=open(input("Enter file name: "),"r")

lines=file.readlines()

for line in range(len(lines)):

if line==len(lines)-1:

print("{}".format(lines[line].strip()))

else:

print("{}".format(lines[line].strip()),end="; ")

**Sample Input and Sample Output:**

Enter file name: file.txt

[abcd@gmail.com; pqrs@gmail.com; google@gmail.com; gmail@gmail.com](mailto:abcd@gmail.com;%20pqrs@gmail.com;%20google@gmail.com;%20gmail@gmail.com)

**9.2) Write a program that reads a list of temperatures from a file called temps.txt, converts those temperatures to Fahrenheit, and writes the results to a file called ftemps.txt**

**Program:**

f1=open("temps.txt",'r')

lines=f1.readlines()

f2=open("ftemps.txt",'w')

for i in range(len(lines)):

c=lines[i].strip()

f=round((float(c)\*1.8)+32,2)

f2.write(str(f)+"\n")

f2.close()

**Sample Input:**

45.7

44.5

44.8

43.3

43.2

44.1

**Sample Output:**

114.26

112.1

112.64

109.94

108.14

111.38

**9.3) Write a program to count frequency of characters in a given file.**

**Program:**

with open('example.txt','r') as f1:

data=[]

a=f1.readlines()

for i in a:

data.append(i[:-1])

d={}

for i in data:

for j in i:

if j in d.keys():

d[j]+=1

else:

d[j]=1

print(d)

**Sample Input and Sample Output:**

{'H': 4, 'e': 4, 'a': 4, 't': 2, ' ': 8, 'w': 1, 'v': 1, 'o': 8, 'l': 3, 'i': 6, 'd': 3, 'y': 1, 's': 4, 'B': 1, 'u': 1, 'n': 6, 'c': 1, ',': 2, 'G': 2, 'M': 2, 'r': 2, 'g': 2}

**Experiment-10:**

**10.1) Write a class called Product. The class should have fields called name, amount, and price, holding the product’s name, the number of items of that product in stock, and the regular price of the product. There should be a method get\_price that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called make\_purchase that receives the number of items to be bought and decreases amount by that much.**

**Program:**

class product:

def \_\_init\_\_(self,name,items,price):

self.name=name

self.items=items

self.price=price

def getprice(self,n):

if(n<10):

print("Regular price is charged")

cost=n\*self.price

print("10% discount on above 9 items, 20% discount on 99 items")

elif n>=10 and n<100:

print("10% discount is applied")

cost=n\*self.price

discount=(cost\*10)/100

total=cost-discount

print("Actual cost:",cost)

print("Cost after discount:",total)

print("20% discount on above 99 items")

else:

print("20% discount is applied")

cost=n\*self.price

discount=(cost\*10)/100

total=cost-discount

print("Actual cost:",cost)

print("Cost after discount:",total)

def my\_purchase(self,n):

if(n<10):

print("Regular price is charged")

cost=n\*self.price

print("Final cost:",cost)

elif n>=10 and n<100:

print("10% discount is applied")

cost=n\*self.price

discount=(cost\*10)/100

total=cost-discount

print("Actual cost:",cost)

print("Cost after discount:",total)

else:

print("20% discount is applied")

cost=n\*self.price

discount=(cost\*10)/100

total=cost-discount

print("Actual cost:",cost)

print("Cost after discount:",total)

p=product("PEN",200,5)

n=int(input())

p.getprice(n)

n=int(input())

p.my\_purchase(n)

**Sample Input and Sample Output:**

9

Regular price is charged

10% discount on above 9 items, 20% discount on 99 items

20

10% discount is applied

Actual cost: 100

Cost after discount: 90.0

**10.2) Write a class called Time whose only field is a time in seconds. It should have a method called convert\_to\_minutes that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called convert\_to\_hours that returns a string of hours, minutes, and seconds formatted analogously to the previous method.**

**Program:**

class Time:

def \_\_init\_\_(self,sec):

self.sec=sec

def con\_min(self):

n=self.sec

minu=n//60

seco=n%60

return(str(minu)+":"+str(seco))

def con\_hour(self):

n=self.sec

hour=n//3600

minu=(n//60)%60

seco=n%60

return(str(hour)+":"+str(minu)+":"+str(seco))

a=int(input())

a=Time(a)

print(a.con\_min())

print(a.con\_hour())

**Sample Input and Sample Output:**

120

2:0

0:2:0

**10.3) Write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, c = Converter(9,'inches'). The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the Converter object created above, the user could call c.feet() and should get 0.75 as the result.**

**Program:**

class converter:

def \_\_init\_\_(self,length,unit):

self.length=length

self.unit=unit

def feet(self):

if(self.unit=='feet'):

return self.length

elif(self.unit=='inches'):

return self.length/12

elif(self.unit=='yards'):

return self.length/0.333

elif(self.unit=='miles'):

return self.length/0.000189

elif(self.unit=='millimeters'):

return self.length/304.8

elif(self.unit=='centimeters'):

return self.length/30.48

elif(self.unit=='meters'):

return self.length/0.305

elif(self.unit=='kilometers'):

return self.length/0.000305

def inches(self):

if(self.unit=='feet'):

return self.length\*12

elif(self.unit=='inches'):

return self.length

elif(self.unit=='yards'):

return self.length\*36

elif(self.unit=='miles'):

return self.length\*63360

elif(self.unit=='millimeters'):

return self.length\*0.0393701

elif(self.unit=='centimeters'):

return self.length\*0.393701

elif(self.unit=='meters'):

return self.length\*39.3701

elif(self.unit=='kilometers'):

return self.length\*39370.1

def yards(self):

if(self.unit=='feet'):

return self.length\*0.333333

elif(self.unit=='inches'):

return self.length\*0.0277778

elif(self.unit=='yards'):

return self.length

elif(self.unit=='miles'):

return self.length\*1760

elif(self.unit=='millimeters'):

return self.length\*0.00109361

elif(self.unit=='centimeters'):

return self.length\*0.0109361

elif(self.unit=='meters'):

return self.length\*1.09361

elif(self.unit=='kilometers'):

return self.length\*1093.61

def miles(self):

if(self.unit=='feet'):

return self.length\*0.000189394

elif(self.unit=='inches'):

return self.length\*63360

elif(self.unit=='yards'):

return self.length\*0.027777728

elif(self.unit=='miles'):

return self.length

elif(self.unit=='millimeters'):

return self.length/1609344

elif(self.unit=='centimeters'):

return self.length/160934.4

elif(self.unit=='meters'):

return self.length/1609.344

elif(self.unit=='kilometers'):

return self.length/1.609

def kilometers(self):

if(self.unit=='feet'):

return self.length/3280.84

elif(self.unit=='inches'):

return self.length/39370.1

elif(self.unit=='yards'):

return self.length/1093.61

elif(self.unit=='miles'):

return self.length/0.621371

elif(self.unit=='millimeters'):

return self.length/1000000

elif(self.unit=='centimeters'):

return self.length/100000

elif(self.unit=='meters'):

return self.length/1000

elif(self.unit=='kilometers'):

return self.length

def meters(self):

if(self.unit=='feet'):

return self.length/3.28084

elif(self.unit=='inches'):

return self.length/39.3701

elif(self.unit=='yards'):

return self.length/1.09361

elif(self.unit=='miles'):

return self.length/0.000621371

elif(self.unit=='millimeters'):

return self.length/1000

elif(self.unit=='centimeters'):

return self.length/100

elif(self.unit=='meters'):

return self.length

elif(self.unit=='kilometers'):

return self.length/0.001

def centimeters(self):

if(self.unit=='feet'):

return self.length/0.0328084

elif(self.unit=='inches'):

return self.length/0.393701

elif(self.unit=='yards'):

return self.length/0.0109361

elif(self.unit=='miles'):

return self.length\*160934

elif(self.unit=='millimeters'):

return self.length/10

elif(self.unit=='centimeters'):

return self.length

elif(self.unit=='meters'):

return self.length\*100

elif(self.unit=='kilometers'):

return self.length\*100000

def millimeters(self):

if(self.unit=='feet'):

return self.length\*304.8

elif(self.unit=='inches'):

return self.length/0.0393701

elif(self.unit=='yards'):

return self.length/0.00109361

elif(self.unit=='miles'):

return self.length\*1609340

elif(self.unit=='millimeters'):

return self.length

elif(self.unit=='centimeters'):

return self.length\*10

elif(self.unit=='meters'):

return self.length\*100

elif(self.unit=='kilometers'):

return self.length\*1000000

len=float(input())

type=input()

c=converter(len,type)

print("Length in feet:",round(c.feet(),3))

print("Length in inches:",round(c.inches(),3))

print("Length in yards:",round(c.yards(),3))

print("Length in miles:",round(c.miles(),3))

print("Length in kilometers:",round(c.kilometers(),3))

print("Length in meters:",round(c.meters(),3))

print("Length in centimeters:",round(c.centimeters(),3))

print("Length in millimeters:",round(c.millimeters(),3))

**Sample Input and Sample Output:**

9

inches

Length in feet: 0.75

Length in inches: 9.0

Length in yards: 0.25

Length in miles: 570240.0

Length in kilometers: 0.0

Length in meters: 0.229

Length in centimeters: 22.86

Length in millimeters: 228.6

**Experiment-11:**

**11.1) Write a Python class to implement pow(x, n)**

**Program:**

class power:

def pow(self,x,n):

print(x\*\*n)

p=power()

x=int(input())

n=int(input())

p.pow(x,n)

**Sample Input and Sample Output:**

6

2

36

**11.2) Write a Python class to reverse a string word by word.**

**Program:**

class reverse:

def rev\_sent(self,sentence):

words=sentence.split()

reverse\_sentence=" ".join(reversed(words))

print(reverse\_sentence)

c=reverse()

c.rev\_sent(input())

**Sample Input and Sample Output:**

Technical Hub

Hub Technical

**Experiment-12:**

**12.1) Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox.**

**Program:**

from tkinter import filedialog

from tkinter import Tk

from tkinter import \*

root=Tk()

root.fileName=filedialog.askopenfilename(filetypes=(("Text Files",".txt"),("All Files","\*.\*")))

text1=open(root.fileName).read()

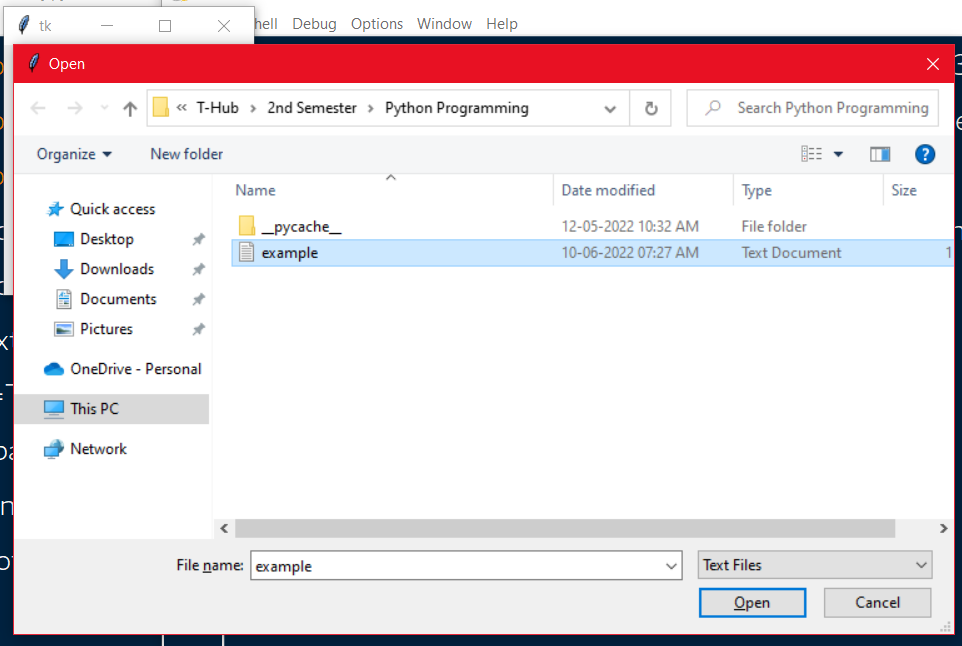
T=Text(root,height=25,width=80)

T.pack()

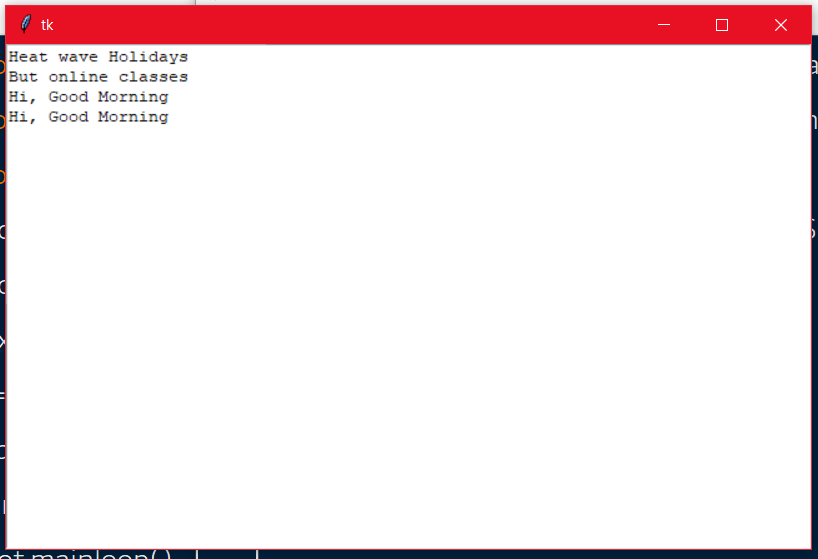
T.insert(END,text1)

root.mainloop()

**Sample Input:**



**Sample Output:**



**12.2) Write a program to demonstrate Try/except/else.**

**Program:**

try:

a=int(input())

b=int(input())

c=a/b

except ZeroDivisionError:

print("Division Cannot be Possible(b==0)")

else:

print("a/b value is",c)

**Sample Input and Sample Output:**

5

0

Division Cannot be Possible(b==0)

Sample Input and Sample Output:

5

2

a/b value is 2.5

**12.3) Write a program to demonstrate try/finally and with/as.**

**Program:**

file=open("file1.txt",'w')

try:

file.write("Hello students, how are you guys?")

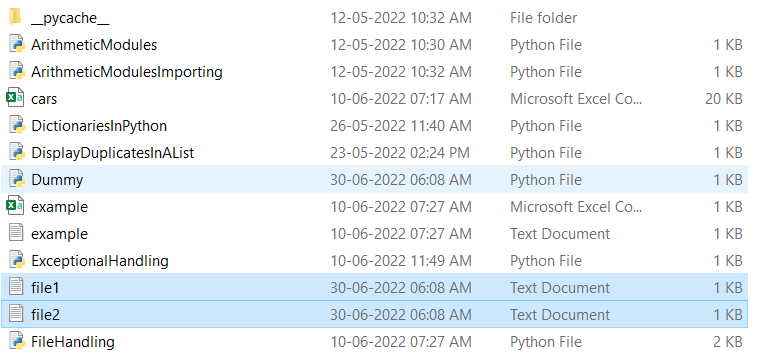
finally:

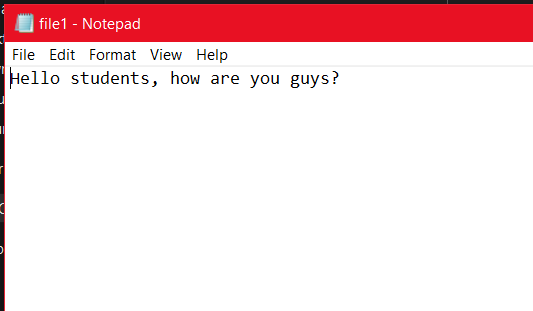
file.close()

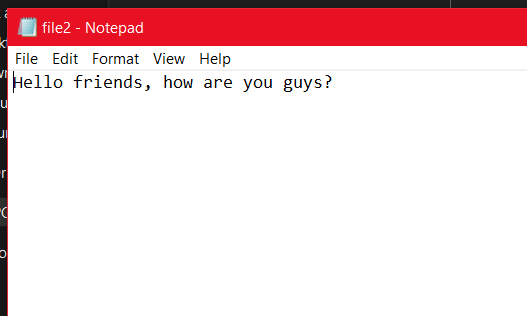
with open("file2.txt",'w') as file:

file.write("Hello friends, how are you guys?")

**Sample Output:**







**List of AUGMENTED EXPERIMENTS:**

**(Minimum of 2 experiments have to be performed)**

**Experiment-15:**

**15) Write a program to find the greatest number that can be formed by using given set of numbers.**

**Program:**

def findMaxNum(arr,n) :

arr.sort(reverse=True)

num=arr[0]

for i in range(1,n):

num=(num\*10)+arr[i]

return num

arr=list(map(int,input().split()))

print(findMaxNum(arr,len(arr)))

**Sample Input and Sample Output:**

8 9 4 5 6 1

986541

**Experiment-16:**

**16) Write a program to find sum of digits of a number till you get single digit sum.**

**Program:**

n=int(input())

while(len(str(n))!=1):

s=0

while(n):

s+=(n%10)

n//=10

n=s

print(n)

**Sample Input and Sample Output:**

4683

3