

Computer Science XI

Practical File 2021-22

Made by-

Ujjwal Kakar

XI – F

Roll no. 39

Program

Mean Average (Multiple Forms

**Source Code-**

def avg(n1,n2,n3,n4):

av=(n1+n2+n3+n4)/4

print(av)

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

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

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

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

avg(a,b,c,d)

def avg2(n1,n2,n3,n4):

s=0

l=4

for k in n1,n2,n3,n4:

if k=="":

k=0

l-=1

s+=float(k)

av=(s)/l

print(av)

a=input("Enter a number: ")

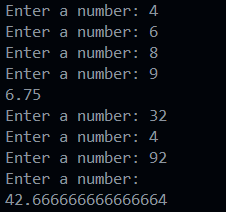
b=input("Enter a number: ")

c=input("Enter a number: ")

d=input("Enter a number: ")

avg2(a,b,c,d)

**Output-**



Program

Cubing (Multiple Forms)

**Source Code-**

#23/3/22

#Method 1:

def cube(n):

a=n\*n\*n

return a

num=float(input("Enter a number: "))

print(cube(num))

#Method 2:

def cube2(n): return n\*\*3

num2 = float(input("Enter a Number: "))

print(cube2(num2))

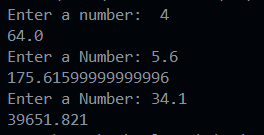
#Method 3:

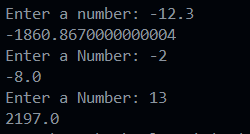
cube3 = lambda n : n\*\*3

num3 = float(input("Enter a Number: "))

print(cube3(num3))

**Output-**





Program

Area of Circle (Multiple Forms)

**Source Code-**

#25/3/22

from math import \*

#Method 1:

def ar(rad):

a=pi\*rad\*rad

return a

r=float(input("Enter the radius: "))

print("The area is: ",ar(r))

#Method 2:

def ar2(rad):

return pi\*(rad\*\*2)

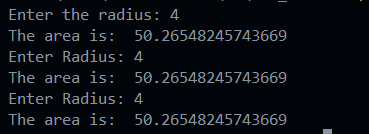
print("The area is: ",ar2(float(input("Enter Radius: "))))

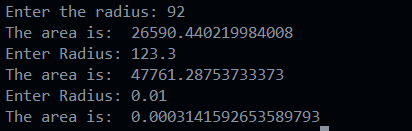
#Method 3:

ar3 = lambda rad : pi\*(r\*\*2)

print("The area is: ",ar3(float(input("Enter Radius: "))))

**Output-**





Program

Calculator

**Source Code-**

#28/3/22

from math import \*

def conNum(x):

if float(x) == round(float(x)):

return round(float(x))

elif float(x) != round(float(x)):

return round(float(x),3)

def simpleCalc():

print("Welcome to Simple Calculator, + adds , - subtracts , x multiplies , / divides , ^ exponents, exit by typing 'out' in operation")

cv = conNum(input("Enter number: "))

while True:

fun = input("Enter operation: ")

if (fun == "+"):

cv += conNum(input("Enter number: "))

print(cv)

if (fun == "-"):

cv -= conNum(input("Enter number: "))

print(cv)

if (fun.lower() in ("x","\*") ):

cv \*= conNum(input("Enter number: "))

print(cv)

if (fun == "/"):

cv /= conNum(input("Enter number: "))

print(cv)

if (fun == "^"):

v = conNum(input("Enter number: "))

cv=pow(cv,v)

print(cv)

if (fun.lower() == "out"):

break

def VolumeCalc():

print("Welcome to Volume Calculator, write shape name to find volume ")

shapeType = input("Enter Shape (Cube, Cuboid, Cone, Cylinder, Sphere, Hemisphere, Frustrum): ")

if shapeType.lower() == "cube":

print("The Volume of Cube is",

conNum(input("Enter Side: "))\*\*3)

if shapeType.lower() == "cuboid":

print("The Volume of Cuboid is",

conNum(input("Enter Length: "))\*conNum(input("Enter Breadth: "))\*conNum(input("Enter Height: ")))

if shapeType.lower() == "cone":

print("The Volume of Cone is",

pi\*(conNum(input("Enter radius: "))\*\*2)\*conNum(input("Enter height: "))/3)

if shapeType.lower() == "cylinder":

print("The Volume of cylinder is",

pi\*(conNum(input("Enter radius: "))\*\*2)\*conNum(input("Enter height: ")))

if shapeType.lower() == "sphere":

print("The Volume of Sphere is",

4\*pi/3\*conNum(input("Enter radius: "))\*\*3)

if shapeType.lower() == "hemisphere":

print("The Volume of Hemisphere is",

2\*pi/3\*conNum(input("Enter radius: "))\*\*3)

if shapeType.lower() == "frustrum":

print("The Volume of Frustrum is",fabs(

(conNum(input("Enter radius 1: "))\*\*3 - conNum(input("Enter radius 2: "))\*\*3) \* pi / 3 \* conNum(input("Enter height: ")) ))

def SAcalc():

print("Welcome to Surface Area Calculator, write shape name to find Surface Area ")

shapeType = input("Enter Shape (Cube, Cuboid, Cone, Cylinder, Sphere, Hemisphere, Frustrum): ")

if shapeType.lower() == "cube":

print("The Surface Area of Cube is",

6\*conNum(input("Enter Side: "))\*\*2)

if shapeType.lower() == "cuboid":

l=conNum(input("Enter Length: "))

b=conNum(input("Enter Breadth: "))

h=conNum(input("Enter Height: "))

print("The Surface Area of Cuboid is",2\*(l\*b + b\*h + h\*l))

if shapeType.lower() == "cone":

r = conNum(input("Enter radius: "))

h = conNum(input("Enter height: "))

print("The Surface Area of Cone is",

pi \* r \* (r + (r\*\*2 + h\*\*2)\*\*(1/2) ))

if shapeType.lower() == "cylinder":

r = conNum(input("Enter radius: "))

h = conNum(input("Enter height: "))

print("The Surface Area of cylinder is",

2\*pi\*r\*(r+h))

if shapeType.lower() == "sphere":

print("The Surface Area of Sphere is",

4\*pi\*conNum(input("Enter radius: "))\*\*2)

if shapeType.lower() == "hemisphere":

print("The Surface Area of Hemisphere is",

3\*pi\*conNum(input("Enter radius: "))\*\*2)

if shapeType.lower() == "frustrum":

print("The Surface Area of Frustrum is ",

(conNum(input("Enter radius 1: ")) + conNum(input("Enter radius 2: "))) \* pi \* (conNum(input("Enter height: ")) + 2) )

def TrigCalc():

a = lambda x,y: x+" of "+str(y)+" is "+str(eval(x+"("+str(y)+")"))

print("Welcome to trignometric calculator, What operation would you like to perform?")

trigOp = input("Enter function(sin,cos,tan,asin,acos,atan): ").lower()

if trigOp == "sin": print(a(trigOp , float(input("Enter Value: "))))

elif trigOp == "cos": print(a(trigOp , float(input("Enter Value: "))))

elif trigOp == "tan": print(a(trigOp , float(input("Enter Value: "))))

elif trigOp == "asin": print(a(trigOp , float(input("Enter Value: "))))

elif trigOp == "acos": print(a(trigOp , float(input("Enter Value: "))))

elif trigOp == "atan": print(a(trigOp , float(input("Enter Value: "))))

def CashCalc():

print("Welcome to Currency convertor: it converts the currencies to different forms")

base = input("Enter a supported currency(USD,EUR,JPY,GBP,AUD,CAD,INR): ").upper()

val = float(input("Enter amount:"))

convertto = input("Enter 2nd supported currency(USD,EUR,JPY,GBP,AUD,CAD,INR): ").upper()

if base=="USD":

if convertto=="USD":print("Exchange Value is",val\*1)

if convertto=="EUR":print("Exchange Value is",val\*0.93)

if convertto=="JPY":print("Exchange Value is",val\*127.71)

if convertto=="GBP":print("Exchange Value is",val\*0.78)

if convertto=="AUD":print("Exchange Value is",val\*1.4)

if convertto=="CAD":print("Exchange Value is",val\*1.27)

if convertto=="INR":print("Exchange Value is",val\*76.69)

if base=="EUR":

if convertto=="USD":print("Exchange Value is",val\*1.07)

if convertto=="EUR":print("Exchange Value is",val\*1)

if convertto=="JPY":print("Exchange Value is",val\*136.97)

if convertto=="GBP":print("Exchange Value is",val\*0.84)

if convertto=="AUD":print("Exchange Value is",val\*1.5)

if convertto=="CAD":print("Exchange Value is",val\*1.37)

if convertto=="INR":print("Exchange Value is",val\*82.17)

if base=="JPY":

if convertto=="USD":print("Exchange Value is",val\*0.0078)

if convertto=="EUR":print("Exchange Value is",val\*0.0073)

if convertto=="JPY":print("Exchange Value is",val\*1)

if convertto=="GBP":print("Exchange Value is",val\*0.0062)

if convertto=="AUD":print("Exchange Value is",val\*0.011)

if convertto=="CAD":print("Exchange Value is",val\*0.010)

if convertto=="INR":print("Exchange Value is",val\*0.60)

if base=="GBP":

if convertto=="USD":print("Exchange Value is",val\*1.27)

if convertto=="EUR":print("Exchange Value is",val\*1.19)

if convertto=="JPY":print("Exchange Value is",val\*162.54)

if convertto=="GBP":print("Exchange Value is",val\*1)

if convertto=="AUD":print("Exchange Value is",val\*1.78)

if convertto=="CAD":print("Exchange Value is",val\*1.62)

if convertto=="INR":print("Exchange Value is",val\*97.62)

if base=="AUD":

if convertto=="USD":print("Exchange Value is",val\*0.71)

if convertto=="EUR":print("Exchange Value is",val\*0.67)

if convertto=="JPY":print("Exchange Value is",val\*91.2)

if convertto=="GBP":print("Exchange Value is",val\*0.56)

if convertto=="AUD":print("Exchange Value is",val\*1)

if convertto=="CAD":print("Exchange Value is",val\*0.91)

if convertto=="INR":print("Exchange Value is",val\*54.77)

if base=="CAD":

if convertto=="USD":print("Exchange Value is",val\*0.78)

if convertto=="EUR":print("Exchange Value is",val\*0.73)

if convertto=="JPY":print("Exchange Value is",val\*100.04)

if convertto=="GBP":print("Exchange Value is",val\*0.62)

if convertto=="AUD":print("Exchange Value is",val\*1.10)

if convertto=="CAD":print("Exchange Value is",val\*1)

if convertto=="INR":print("Exchange Value is",val\*60.08)

if base=="INR":

if convertto=="USD":print("Exchange Value is",val\*0.013)

if convertto=="EUR":print("Exchange Value is",val\*0.012)

if convertto=="JPY":print("Exchange Value is",val\*1.67)

if convertto=="GBP":print("Exchange Value is",val\*0.010)

if convertto=="AUD":print("Exchange Value is",val\*0.018)

if convertto=="CAD":print("Exchange Value is",val\*0.017)

if convertto=="INR":print("Exchange Value is",val\*1)

optionType = 0

print("Welcome to General Calculator, It has many options such as volume, area, trigonometry, currency, general.")

print("Select what you want to use")

while True:

optionType = input("Enter your calculator type:\n 1)Volume\n 2)Surface Area\n 3)Trigonometry\n 4)Currency\n 5)General\n 6)Exit\n\n ")

if optionType.lower() == "volume" or optionType.lower() == "1":

VolumeCalc()

if optionType.lower() == "surface area" or optionType.lower() == "2":

SAcalc()

if optionType.lower() == "trigonometry" or optionType.lower() == "3":

TrigCalc()

if optionType.lower() == "currency" or optionType.lower() == "4":

CashCalc()

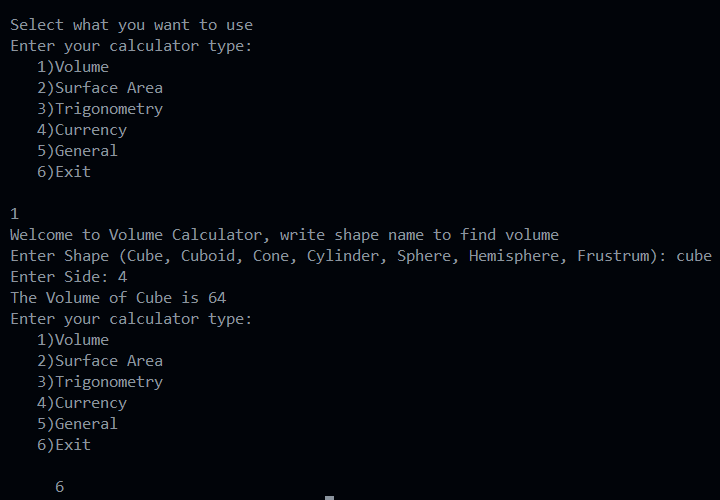
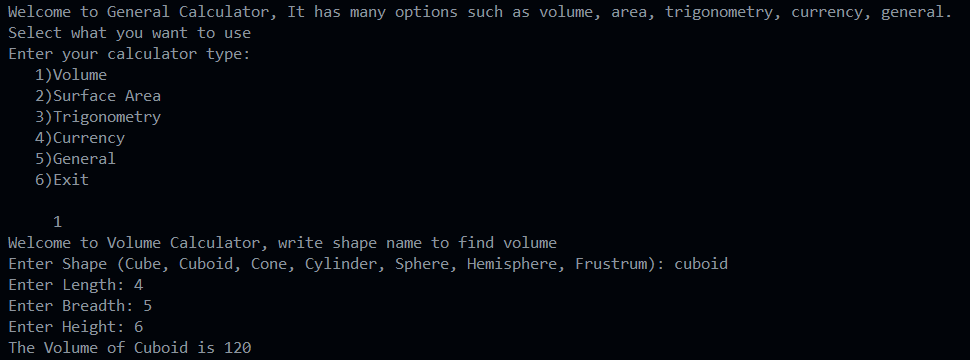
if optionType.lower() == "general" or optionType.lower() == "5":

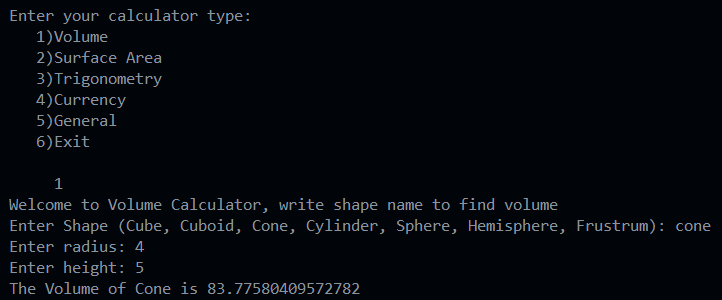
simpleCalc()

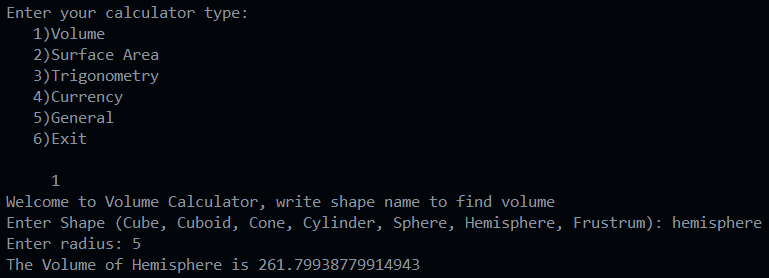
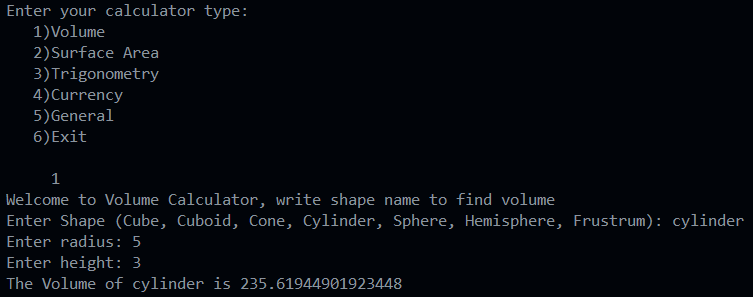
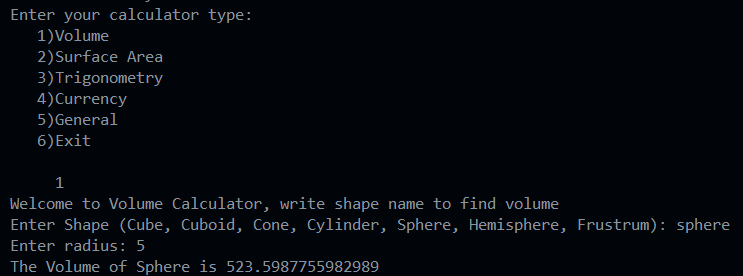
if optionType.lower() == "exit" or optionType.lower() == "6":

break

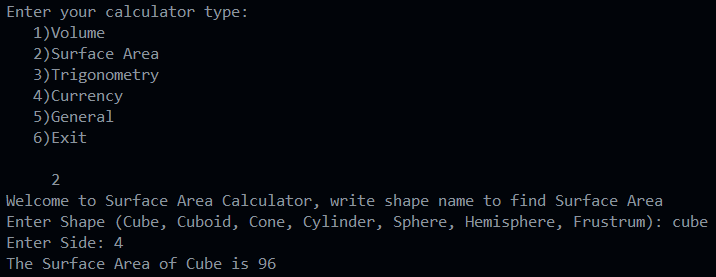
**Output-**

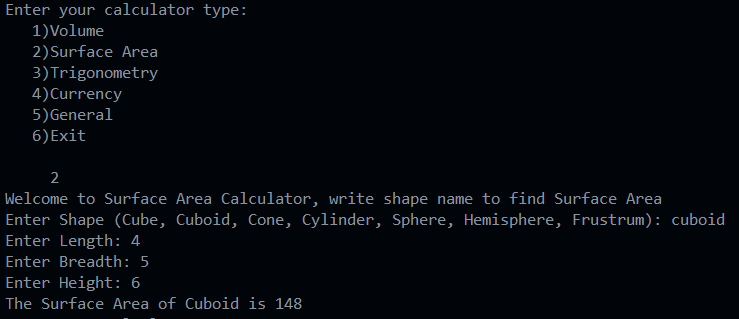


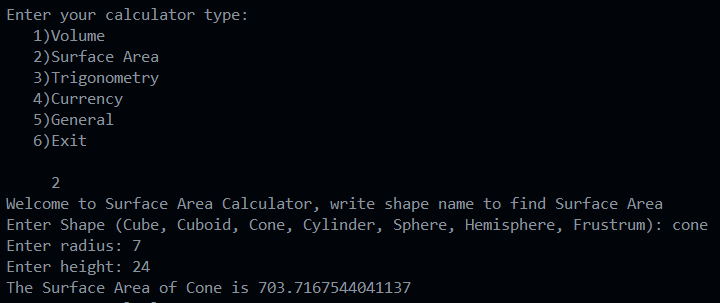


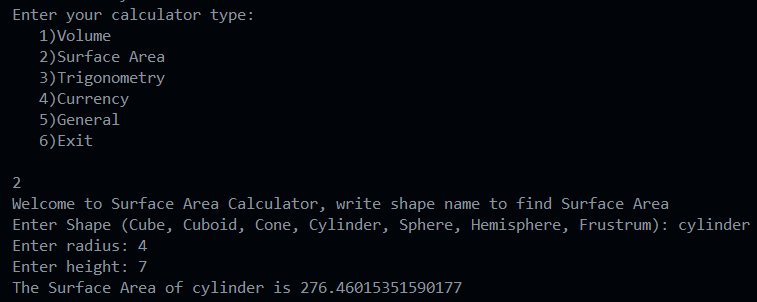


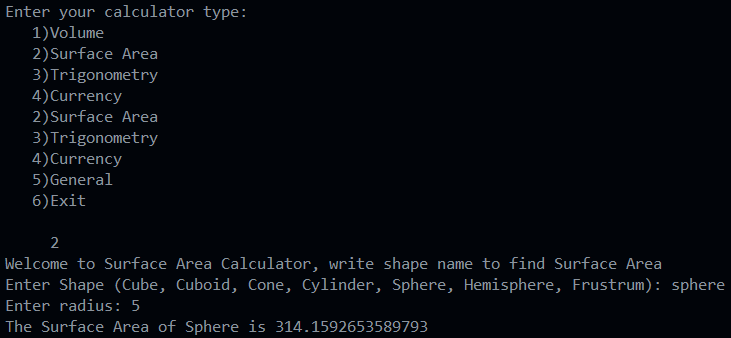


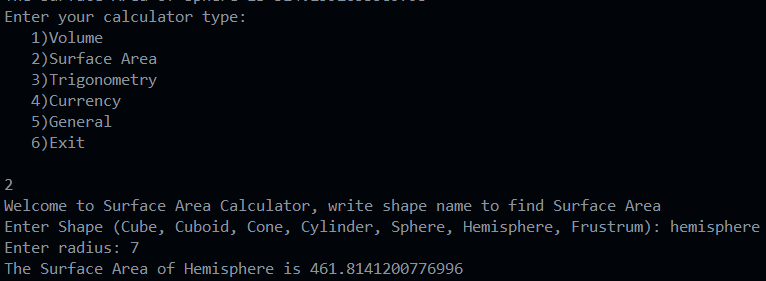












Program

Line Drawing function for parameter usage

**Source Code-**

#4/4/22

def drawline(sym,t=20):

for i in range(t):

print(sym,end='')

print()

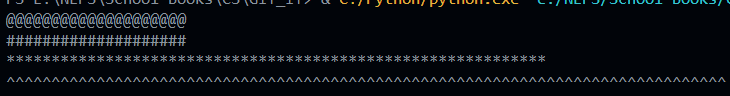
drawline(sym='@')

drawline('#')

drawline('\*',60)

drawline(t=80 ,sym='^')

**Output-**



Program

Random Selection (Multiple Methods)

**Source Code-**

#8/4/22

import random

subj = ["Computer Science", "IP", "Physics", "Maths"]

#Method 1:

print(random.choice(subj))

#Method 2:

print(subj[random.randrange(3)])

#Method 3:

random.shuffle(subj)

print(subj[0])

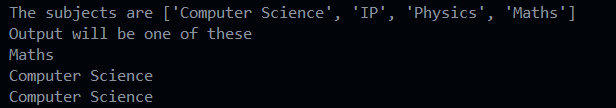
f=['Apple','Mango','Orange','Banana','Avacado']

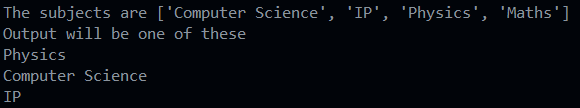
print('Original list: ',f)

random.shuffle(f)

print('Reshuffled list: ',f)

**Output-**







Program

Random 3 digit number and Sum (with Alternate methods)

**Source Code-**

#8/4/22

from random import random,randint

n=(random()\*900+100)//1

print('The number is: ',n)

s = 0

while n >0:

s +=n%10

n//=10

print('The sum of the digits is:',s)

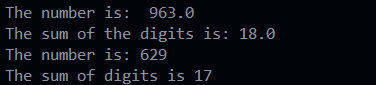
#Alternate Method

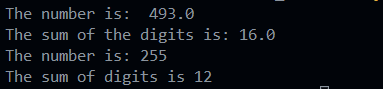
n2 = randint(100,999)

print("The number is:",n2)

print("The sum of digits is",sum([int(k) for k in str(n2)]))

**Output-**





Program

List Fillers (random library application for dummy Data Strucuture Analysis)

**Source Code-**

#11/4/22

from random import randint

def fill\_list(L , num , min , max):

for \_ in range(num): L.append(randint(min,max))

low , up , elem = int(input("Enter Minimum: ")) , int(input("Enter Maximum: ")) , int(input("Enter total terms: "))

a = []

b = []

fill\_list(b,elem,low,up)

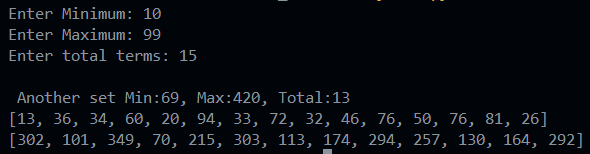
print("\n Another set Min:69, Max:420, Total:13")

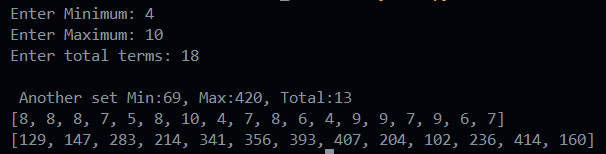
fill\_list(a , min=69 , max=420 , num=13)

print(b)

print(a)

**Output-**





Program

Binary Search Algorithm to reduce time complexity

**Source Code-**

#18/4/22

from random import randint

def bin (l,el):

mid=len(l)//2

low=(0)

high=len(l)-1

passes=0

while l[mid]!=el and low<=high:

if el>l[mid]:

low=mid+1

else:

high=mid-1

mid=(high+low)//2

passes+=1

if low>high:

return None

else:

return mid,passes

a=[]

for \_ in range(12):

a.append(randint(1,100))

a.sort()

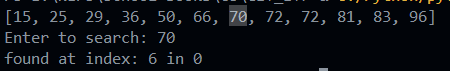
print(a)

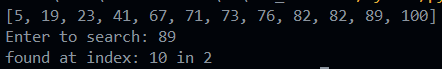
v=int(input('Enter to search: '))

op = bin(a,v)

print("found at index:",op[0],"in",op[1])

**Output-**





Program

Random application (Lottery Generation)

**Source Code-**

#19/4/22

from random import uniform

print("Lottery number is between 1-100")

a = round(uniform(0,100),4)

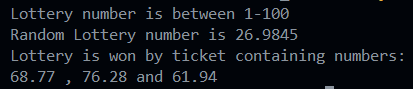
print("Random Lottery number is",a)

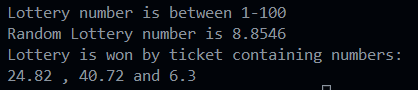
print("Lottery is won by ticket containing numbers: ")

a = lambda : uniform(1,100)//0.01 /100

print(a(),",",a(),"and",a())

**Output-**





Program

Random application (Direction Selector, Treasure map-precursor)

**Source Code-**

#21/4/22

import random

direction = random.choice(["East", "West", "North","South"])

print("Randomly selected cardinal direction is",direction)

def d():

dir = random.choice(["East", "West", "North","South"])

return dir

print("Random non cardinal Direction is",d()+"-"+d())

**Output-**







Program

Vowel Counter (Multiple Methods)

**Source Code-**

#22/4/22

#Method 1:

def countvowel(s):

c=0

for ch in s:

if ch in 'aeiouAEIOU':

c+=1

return c

sin=input('Enter a string: ')

count=countvowel(sin)

print('Total number of vowels in the string are: ',count)

#Method 2:

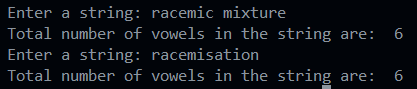
def vowelcount(st): return len(['' for k in st if st.lower() in "aeiou"])

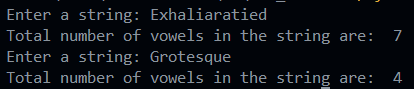
a=input('Enter a string: ')

d=countvowel(a)

print('Total number of vowels in the string are: ',d)

**Output-**





Program

Higher Lower (Simple Guessing Game)

**Source Code-**

#22/4/22

#Higher lower game

import random

initial = round(100000 \* random.random())

while True:

print("number is: ",initial)

print("Higher or lower\n")

new = round(100000 \* random.random())

input()

if new>initial:

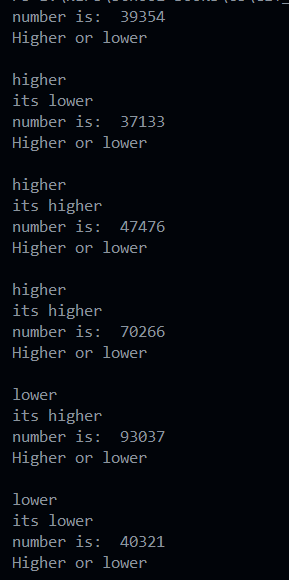
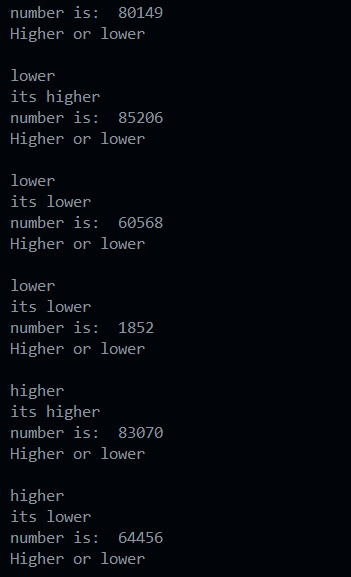
print('its higher')

if new<initial:

print('its lower')

initial = new

**Output-**



Program

Character counter

**Source Code-**

#22/4/22

def countChar(s,ch):

c=0

for i in s:

if i==ch:

c+=1

return c

sin=input('Enter a string: ')

ch1=input('Enter the character to count: ')

f=countChar(sin,ch1)

if f==0:

print(ch1, "does not exist.")

else:

print(ch1,'exist',f,'times.')

#Alernate Method

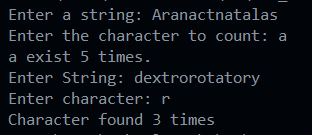
chCount = lambda x,y : len(["" for k in x if k == y])

v = chCount(input("Enter String: "),input("Enter character: "))

if v!=0: print("Character found",v,"times")

else: print("Character not found")

**Output-**



Program

Passing Immutables as parameters

**Source Code-**

#22/4/22

#passing immutable tuple to a function

def ttl(A):

A=list(A)

A[0]=A[0]\*2

A[1]=A[1]+10

print(A)

t=(100,200)

print(t)

ttl(t)

**Output-**



Program

Tuple Sotrage and Multi Value Return Unpacking

**Source Code-**

#22/4/22

def countOddEven(t):

odd = len(['' for k in t if k%2==1])

return odd , len(t) - odd

store = tuple()

times = int(input("Number of elements: "))

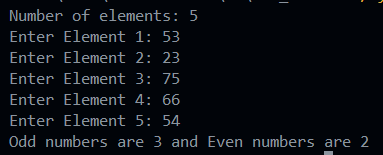
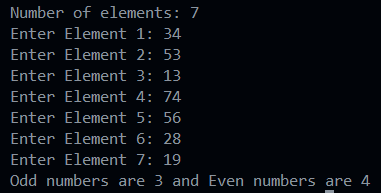
for \_ in range(times):

store+= (int(input("Enter Element "+str(\_+1)+": ")),)

a,b = countOddEven(store)

print("Odd numbers are",a , "and Even numbers are",b)

**Output-**



Program

Functions for Dictionary undates

**Source Code-**

#22/4/22

def marksu(s,nm):

s['Marks']+=nm

s["Status"]="Updated"

s1={'Name':'Akash','Marks':56,'Status':'Old'}

s2={'Name':'Chinmay','Marks':60,'Status':'Old'}

s3={'Name':'Chirag','Marks':50,'Status':'Old'}

print("Original data: ")

print(s1)

print(s2)

print(s3)

marksu(s1,70)

marksu(s2,80)

marksu(s3,75)

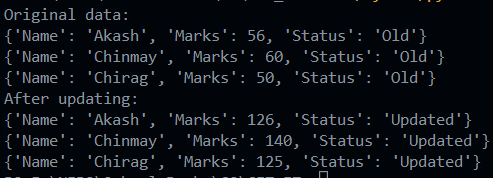
print("After updating: ")

print(s1)

print(s2)

print(s3)

**Output-**



Program

Frequncy Distribution by dictionary purposing

**Source Code-**

#23/4/22

def freq(l):

d={}

for i in l:

if i not in d:

d[i]=1

else:

d[i]+=1

return d

l1=[1,2,31,4,1,4,6,2,1,6,6,2]

print(freq(l1))

**Output-**



Program

**Source Code-**

**Output-**

Program

**Source Code-**

**Output-**

Program

**Source Code-**

**Output-**

Program

**Source Code-**

**Output-**

Program

**Source Code-**

**Output-**

Program

**Source Code-**

**Output-**

Program

**Source Code-**

**Output-**