Roll No: 21BCE528

Aim: a. Develop a python program and make the simple calculator which using and conditional loop.

1. Write a function areaTriangle that takes the lengths of three sides of the triangle as input parameters and returns the area of the triangle as a output. Also, assert that sum of the length of any two sides is greater than the third side. Write a main function that accepts as command line arguments and computers the area of triangle using the function areaTriangle.
2. Write a function that takes two numbers as input parameters and returns True or False depending on whether they are co-primes. Two numbers are said to be co-prime if they do not have any common divisor other than one.
3. Write a function that takes a string as a parameter and returns a string with every successive repetitive character replaced with a star(*). For Example, ‘balloon’ is returned as ‘bal*o\*n’.
4. Write a function that takes a number as n input parameter and returns the correspond text in words, for example, on input 452, the function should return ‘Four Five Two’. Use a dictionary for mapping to digits to their string representation.
5. Write a recursive function that takes number n as an input parameter and print n-digit strictly increasing number.

User Inputs related Programs

1. The bell shaped Gaussian function,

f(x)= 1/(s \* √(2π )) exp〖[ (-1)/2 〖((x-m)/s)〗^2]〗

is one of the most widely used functions in science and technology. The parameters m and s > 0 are prescribed real numbers. Make a program for evaluating this function for different values of s, x and m. Ask user to input the values.

1. A car driver, driving at velocity v0, suddenly puts on the brake. What braking distance d is needed to stop the car? One can derive, using Newton’s second law of motion or a corresponding energy equation, that:

d= 1/2 (v\_0^2)/μg

Make a program for computing d above equation, when the initial car velocity v0 and the friction coefficient µ are given on the command line. Run the program for two cases: v0 = 120 and v0 = 50 km/h, both with µ = 0.3 (µ is dimensionless).

(Note: convert the velocity in m/s)

Python Classes related

1. Define a class Bank that keeps track of bank customers. The class should contain the following data member:

Data member name Details

name

Name of customer accountNum Account Number type

Account Type amount

Amount deposited in the bank account interest

Interest earned by the customer

The class should support the following methods:

1. **init** for initializing the data members.
2. deposit for depositing money in the members.
3. withdrawal for withdrawing money from the account
4. findInterest that determines the interest on the basis of amount in the account

num1 = int(input("Enter Number 1:")) num2 = int(input("Enter Number 2:")) op = input("Enter operator")

if op=='+':

print(num1+num2) elif op=='-':

print(num1-num2) elif op=='\*':

print(num1\*num2) elif op=='/':

print(num1/num2) else:

print("Please Enter correct Operator")

Please Enter correct Operator

#Area of Traingle def Area(a,b,c):

if(a+b>c and a+c>b and b+c>a): s = (a+b+c)/2

area = round((s\*(s-a)\*(s-b)\*(s-c))\*\*0.5,5)

print("Area of Traingle:"+ str(area))

else:

print("Cannot from Triangle")

a = int(input("Enter 1st side:"))

b = int(input("Enter 2st side:"))

c = int(input("Enter 3st side:")) Area(a,b,c)

Area of Traingle:6.0

#coprimes

n1 = int(input("Enter 1st number:")) n2 = int(input("Enter 2nd number:")) def divisor(num):

list=[]

for i in range(2,int(num-1)): if num%i==0:

list.append(i) return list

l1 = divisor(n1) l2 = divisor(n2) print(l1) print(l2)

same = False for i in l1:

for j in l2:

if i==j:

same = True if(same == False):

print("Given numbers are coprime") else:

print("Given numbers are not coprime")

[2, 5] []

Given numbers are coprime

s = input("Enter String") list =[]

s1 = ""

for i in s:

if i not in list: s1+=i

elif i in list: s1+='\*'

list.append(i) print(s1)

bal\*o\*n

def number\_to\_words(number): digits = {

'0': 'zero',

'1': 'one',

'2': 'two',

'3': 'three',

'4': 'four',

'5': 'five',

'6': 'six',

'7': 'seven',

'8': 'eight',

'9': 'nine'

}

word = ''

for digit in str(number): if digit in digits:

word += digits[digit] + ' ' return word.strip()

number = 718

word = number\_to\_words(number) print(word)

seven one eight

speed = int(input("enter speed in km/h:")) speed1 = round(speed \* 5/18,4)

friction = 0.3

g = 9.8

distance = round((1/2)\*((speed1\*\*2)/(friction\*g)),3) print(distance)

1181.027

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

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

m = int(input("Enter m:")) if s>0:

f = round(((1/(s\*((2\*3.14)\*\*0.5)))\*\*((-1/2)\*((x-m)/s)\*\*2)),3) print(f)

1.119

def distance(v0, mu):

v0\_mps = v0 \* 1000 / 3600 g = 9.81

d = 0.5 \* v0\_mps\*\*2 / (mu \* g) return d

v0\_1 = 120

mu\_1 = 0.3

d\_1 = distance(v0\_1, mu\_1)

print(f"Braking distance (d) for v0 = {v0\_1} km/h and mu = {mu\_1} is{d\_1:.2f} meters.") v0\_2 = 50

mu\_2 = 0.3

d\_2 = distance(v0\_2, mu\_2)

print(f"Braking distance (d) for v0 = {v0\_2} km/h and mu = {mu\_2} is{d\_2:.2f} meters.")

Braking distance (d) for v0 = 120 km/h and mu = 0.3 is188.77 meters. Braking distance (d) for v0 = 50 km/h and mu = 0.3 is32.77 meters.

class Bank:

def init (self, name, accountNum, accountType, amount): self.name = name

self.accountNum = accountNum self.accountType = accountType self.amount = amount self.interest = 0

def deposit(self, amount): self.amount += amount

print(f"Deposited {amount} in the account. Current balance:{self.amount}") def withdrawal(self, amount):

if self.amount >= amount: self.amount -= amount

print(f"Withdrawn {amount} from the account. Current balance: {self.amount}") else:

print("Insufficient balance. Withdrawal not allowed.") def findInterest(self):

if self.amount >= 500000: self.interest = self.amount \* 0.08

elif 300000 <= self.amount < 500000: self.interest = self.amount \* 0.07

elif 100000 <= self.amount < 300000: self.interest = self.amount \* 0.05

else:

self.interest = self.amount \* 0.03 print(f"Interest earned: {self.interest:.2f}")

customer1 = Bank("abc", "1111111111", "Savings", 400000)

customer2 = Bank("xyz", "2222222222", "Current", 800000) customer1.deposit(50000)

customer1.findInterest() customer2.withdrawal(100000) customer2.findInterest()

Deposited 50000 in the account. Current balance:450000 Interest earned: 31500.00

Withdrawn 100000 from the account. Current balance: 700000 Interest earned: 56000.00

 