Task : Session 1

### Q1 :- Print the given strings as per stated format.

Given strings:

"Data" "Science" "Mentorship" "Program"

"By" "Wayspire"

Output:

Data-Science-Mentorship-Program-started-By-Wayspire

Concept- [Seperator and End]

# Write your code here

print("Data","Science","Mentorship","Program","By","Wayspire",sep='-',end='\n')

### Q2:- Write a program that will convert celsius value to fahrenheit.

# Write your code here

celsius=int(input("enter the celsius temperature:" ))

farenheit= 9/5\*celsius +32

print("celsuis value in farenheit is",farenheit)

### Q3:- Take 2 numbers as input from the user.Write a program to swap the numbers without using any special python syntax.

# Write your code here

num1= int(input("enter a number:"))

num2= int(input("enter a number:"))

print("Before swapping")

print("first number",num1,"second number",num2)

num3=num1

num1=num2

num2=num3

print("After swapping")

print("first number",num1,"second number",num2)

### Q4:- Write a program to find the euclidean distance between two coordinates.Take both the coordinates from the user as input.

# Write your code here

import math

x1=float(input("enter x coordinte of first point:"))

y1=float(input("enter y coordinate of first point:"))

x2=float(input("enter x coordinte of second point:"))

y2=float(input("enter y coordinate of second point:"))

e=math.sqrt((x2-x1)\*\*2-(y2-y2)\*\*2)

print("euclidean distance between two coordinates",e)

### Q5:- Write a program to find the simple interest when the value of principle,rate of interest and time period is provided by the user.

# Write your code here

P=float(input("enter the principle amount:"))

R=float(input("enter rate of interest:"))

T=int(input("time period in years:"))

SI= (P\*R\*T)/100

print("Simple interest is",SI)

### Q6:- Write a program that will tell the number of dogs and chicken are there when the user will provide the value of total heads and legs.

For example:

Input:

heads -> 4

legs -> 12

Output:

dogs -> 2

chicken -> 2

# Write your code here

total\_heads=int(input("enter number of heads:"))

total\_legs=int(input("enter number of legs:"))

#total\_heads=x+y

#total\_legs=4x+2y

chickens = (4 \* total\_heads - total\_legs) / 2

dogs = total\_heads - chickens

if chickens >= 0 and dogs >= 0:

print("Number of chickens: ", chickens)

print("Number of dogs: ", dogs)

else:

print("No solution exists for given heads and legs")

### Q7:- Write a program to find the sum of squares of first n natural numbers where n will be provided by the user.

# Write your code here

n=int(input("enter a number:"))

sum\_of\_squares=0

if n>0:

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

sum\_of\_squares+=i\*\*2

else:

print("Number should be positive")

print("sum of squares", sum\_of\_squares)

### Q8:- Given the first 2 terms of an Arithmetic Series.Find the Nth term of the series. Assume all inputs are provided by the user.

# Write your code here

n=int(input("enter a number:"))

sum\_of\_squares=0

if n>0:

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

sum\_of\_squares+=i\*\*2

else:

print("Number should be positive")

print("sum of squares", sum\_of\_squares)

### Q9:- Given 2 fractions, find the sum of those 2 fractions.Take the numerator and denominator values of the fractions from the user.

# Write your code here

numerator1=float(input("enter numerator of 1 num:"))

denominator1=float(input("enter denominator of 1 num:"))

numerator2=float(input("enter numerator of 2 num:"))

denominator2=float(input("enter denominator of 2 num:"))

fraction1=numerator1/denominator1

fraction2=numerator2/denominator2

sum= fraction1+ fraction2

print("sum of fractions:",sum)

### Q10:- Given the height, width and breadth of a milk tank, you have to find out how many glasses of milk can be obtained? Assume all the inputs are provided by the user.

Input:<br>

Dimensions of the milk tank<br>

H = 20cm, L = 20cm, B = 20cm

Dimensions of the glass<br>

h = 3cm, r = 1cm

# Write your code here

H=int(input("enter height of milk tank:"))

L=int(input("enter lenght of milk tank:"))

B=int(input("enter breadth of milk tank:"))

vol\_of\_tank= L\*B\*H

h=int(input("enter height of glass:"))

r=int(input("enter radius of glass:"))

vol\_of\_glass= 3.14\*r\*r\*h

n=vol\_of\_tank/vol\_of\_glass

print("number of glasses of milk that can be obtained",n)

### Problem 1: Write a program that will give you in hand monthly salary after deduction on CTC - HRA(10%), DA(5%), PF(3%) and taxes deduction as below:

> Salary(Lakhs) : Tax(%)

\* Below 5 : 0%

\* 5-10 : 10%

\* 10-20 : 20%

\* aboove 20 : 30%

# Write code here

salary=int(input("enter the salary:"))

if salary<50000:

print("no tax")

print("In hand salary",salary)

elif salary< 1000000:

tax= salary \* 10/100

IHS=salary-tax

print("In hand salary",IHS)

elif salary< 2000000:

tax=salary\*20/100

IHS=salary-tax

print("In hand salary",IHS)

elif salary<3000000:

tax=salary\*30/100

IHS=salary-tax

print("In hand salary",IHS)

else:

tax= salary\*30/100

IHS=salary-tax

print("In hand salary",IHS)

### Problem 2: Write a program that take a user input of three angles and will find out whether it can form a triangle or not.

# Write code here

angle1= int(input("enter the first angle "))

angle2= int(input("enter the second angle "))

angle3= int(input("enter the third angle "))

if angle1+angle2+angle3==180 :

print("these angles can form triangle")

else:

print("triangle cannot be formed")

### Problem 3: Write a program that will take user input of cost price and selling price and determines whether its a loss or a profit.

# Write code here

cost\_price=int(input("enter the buying price of item"))

selling\_price=int(input("enter the selling price of item"))

if cost\_price > selling\_price:

print("It is a loss")

else:

print("It is profit ")

### Problem 4: Write a menu-driven program -

1. cm to ft

2. km to miles

3. USD to INR

4. exit

# Write code here

choice = int(input("enter the choice(1-4):"))

if choice == 1:

cm = int(input("enter the value in cm:"))

ft = 1/30 \* cm

print("value in feet is:", ft)

elif choice == 2:

km = int(input("enter the value in km:"))

mile = 0.621371 \* km

print("value in miles is:", mile)

elif choice == 3:

usd = float(input("enter the value in usd:"))

inr = 84 \* usd

print("value in inr is:", inr)

elif choice == 4:

print("exiting the program")

else:

print("no such choice exists")

### Problem 5 - Exercise 12: Display Fibonacci series up to 10 terms.

Note: The Fibonacci Sequence is a series of numbers. The next number is found by adding up the two numbers before it. The first two numbers are 0 and 1. For example, 0, 1, 1, 2, 3, 5, 8, 13, 21. The next number in this series above is 13+21 = 34

# Write code here

n1, n2 = 0, 1

print("Fibonacci series up to 10 terms:")

for i in range(10):

print(n1, end=" ")

nth = n1 + n2

n1, n2 = n2, nth

### Problem 6 - Find the factorial of a given number.

Write a program to use the loop to find the factorial of a given number.

The factorial (symbol: !) means to multiply all whole numbers from the chosen number down to 1.

For example: calculate the factorial of 5

5! = 5 × 4 × 3 × 2 × 1 = 120

Output:120

# Write code here

n=int(input("enter a number:"))

fact=1

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

fact= fact\*i

print(fact)

### Problem 7 - Reverse a given integer number.

Example:

Input:

76542

Output:

24567

# Write code here

def reverse\_integer(number):

reversed\_number = int(str(number)[::-1])

return reversed\_number

# Input

number = int(input("Enter an integer: "))

print("Reversed number:", reverse\_integer(number))

### Problem 8: Take a user input as integer N. Find out the sum from 1 to N. If any number if divisible by 5, then skip that number. And if the sum is greater than 300, don't need to calculate the sum further more. Print the final result. And don't use for loop to solve this problem.

Example 1:

Input:

30

Output:

276

# Write code here

def calculate\_sum(n, current\_sum=0):

# Base case: Stop recursion if n <= 0 or the sum exceeds 300

if n <= 0 or current\_sum > 300:

return current\_sum

# Add current number to the sum if it is not divisible by 5

if n % 5 != 0:

current\_sum += n

# Recursively calculate for the next number

return calculate\_sum(n - 1, current\_sum)

# Input

n = int(input("Enter an integer N: "))

# Calculate the sum and print the result

result = calculate\_sum(n)

print("Final result:", result)

### Problem 9: Write a program that keeps on accepting a number from the user until the user enters Zero. Display the sum and average of all the numbers.

# Write code here

count = 0

sum = 0

while True:

n = int(input("enter a number (enter 0 to stop):"))

if n == 0:

break

count += 1

sum += n

if count != 0:

average = sum / count

print("sum of all numbers:", sum)

print("average of numbers:", average)

else:

print("No numbers were entered")

###Problem 9: Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

# Write code here

lst=[]

for i in range(1999,3201):

if i%7==0 and i%5!=0:

lst.append(i)

print(lst)

###Problem 10: Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. The numbers obtained should be printed in a space-separated sequence on a single line.

# Write code here

lst=[]

for i in range(1000,3000):

digit=[int(digit) for digit in str(i)]

if all(d %2 == 0 for d in digit):

lst.append(i)

print(lst)

###Problem 11: A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with a given steps.

The trace of robot movement is shown as the following:

UP 5

DOWN 3

LEFT 3

RIGHT 2

!

> The numbers after the direction are steps.

> ! means robot stop there.

Please write a program to compute the distance from current position after a sequence of movement and original point.

If the distance is a float, then just print the nearest integer.

Example:

Input:

UP 5

DOWN 3

LEFT 3

RIGHT 2

!

Output:

2

# Write code here

import math

def robot\_distance():

x, y = 0, 0 # Initialize robot at origin

print("Enter commands (end with '!'):")

while True:

command = input()

if command == "!":

break # Stop when "!" is encountered

# Parse the direction and steps

try:

direction, steps = command.split()

steps = int(steps)

# Update position based on direction

if direction == "UP":

y += steps

elif direction == "DOWN":

y -= steps

elif direction == "LEFT":

x -= steps

elif direction == "RIGHT":

x += steps

else:

print("Invalid direction. Use UP, DOWN, LEFT, or RIGHT.")

except ValueError:

print("Invalid command format. Use 'DIRECTION STEPS'.")

# Calculate Euclidean distance

distance = math.sqrt(x\*\*2 + y\*\*2)

# Print the nearest integer value of the distance

print(int(round(distance)))

# Run the program

robot\_distance()

###Problem 12:Write a program to print whether a given number is a prime number or not

# Write code here

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

if n <= 1:

print("Number is not prime")

else:

is\_prime = True

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

if n is\_prime = False

break

if is\_prime:

print("Number is prime")

else:

print("Number is not prime")

###Problem 13:Print all the Armstrong numbers in a given range.

Range will be provided by the user<br>

Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

# Write code here

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

num\_str = str(num)

num\_digits = len(num\_str)

sum\_of\_powers = 0

temp\_num = num

while temp\_num > 0:

digit = temp\_num % 10

sum\_of\_powers += digit \*\* num\_digits

temp\_num //= 10

if sum\_of\_powers == num:

print(f"{num} is an Armstrong number.")

else:

print(f"{num} is not an Armstrong number.")

###Problem 14:Calculate the angle between the hour hand and minute hand.

Note: There can be two angles between hands; we need to print a minimum of two. Also, we need to print the floor of the final result angle. For example, if the final angle is 10.61, we need to print 10.

Input:<br>

H = 9 , M = 0<br>

Output:<br>

90<br>

Explanation:<br>

The minimum angle between hour and minute

hand when the time is 9 is 90 degress.

# Write code here

import math

def clock\_angle(hour, minute):

# Normalize the hour to a 12-hour format

hour = hour % 12

hour\_angle = (hour \* 30) + (minute \* 0.5) # 30 degrees per hour + 0.5 per minute

minute\_angle = minute \* 6 # 6 degrees per minute

angle = abs(hour\_angle - minute\_angle)

min\_angle = min(angle, 360 - angle)

return math.floor(min\_angle)

H = int(input("Enter the hour (H): "))

M = int(input("Enter the minute (M): "))

result = clock\_angle(H, M)

print(result)

###Problem 15:Given two rectangles, find if the given two rectangles overlap or not. A rectangle is denoted by providing the x and y coordinates of two points: the left top corner and the right bottom corner of the rectangle. Two rectangles sharing a side are considered overlapping. (L1 and R1 are the extreme points of the first rectangle and L2 and R2 are the extreme points of the second rectangle).

Note: It may be assumed that the rectangles are parallel to the coordinate axis.

<img src='https://www.geeksforgeeks.org/wp-content/uploads/rectanglesOverlap.png' width='300' height='200'>

# Write code here

def do\_rectangles\_overlap(L1, R1, L2, R2):

"""

Determine if two rectangles overlap.

:param L1: Tuple (x1, y1) - Top-left corner of the first rectangle

:param R1: Tuple (x2, y2) - Bottom-right corner of the first rectangle

:param L2: Tuple (x3, y3) - Top-left corner of the second rectangle

:param R2: Tuple (x4, y4) - Bottom-right corner of the second rectangle

:return: Boolean - True if rectangles overlap, False otherwise

"""

# Check for non-overlapping conditions

if (R1[0] < L2[0] or R2[0] < L1[0] or # One rectangle is to the left of the other

R1[1] > L2[1] or R2[1] > L1[1]): # One rectangle is above the other

return False

return True

# Input

L1 = tuple(map(int, input("Enter top-left corner of Rectangle 1 (x1, y1): ").split()))

R1 = tuple(map(int, input("Enter bottom-right corner of Rectangle 1 (x2, y2): ").split()))

L2 = tuple(map(int, input("Enter top-left corner of Rectangle 2 (x3, y3): ").split()))

R2 = tuple(map(int, input("Enter bottom-right corner of Rectangle 2 (x4, y4): ").split()))

if do\_rectangles\_overlap(L1, R1, L2, R2):

print("Rectangles overlap.")

else:

print("Rectangles do not overlap.")