Q.Find the nos. upto 100 which is divisible by 3 but not by 9.

for m in range(1,100):

    if m % 3==0 and m % 9!=0:

        print(m)

**Output:**

3

6

12

15

21

24

30

33

39

42

48

51

57

60

66

69

75

78

84

87

93

96

Q. for i in range (2,5):

    print("Manu"\*i)

**Output:**

ManuManu

ManuManuManu

ManuManuManuManu

Q. for i in range (15):

    print(i\*2)

**Output:**

0

2

4

6

8

10

12

14

16

18

20

22

24

26

28

Q. def add\_numbers(num1,num2):

    sum=num1+num2

    print("Sum:",sum)

add\_numbers(5,8)

**Output:**

Sum: 13

Q. number=1

while number<200:

    print (number)

    number=number\*2

**Output:**

1

2

4

8

16

32

64

128

Q.string=abcdabccdaaeffaa replace of a with x.

string="abcdabccdaaeffaa"

new\_string=string.replace("a","x")

print(new\_string)

**Output:**

Xbcdxbccdxxeffxx

Q. Display 1 to 50 no.bt after every 5th digit there should be hello msg.

for i in range(1, 50):

if i % 5 == 0:

print("Hello")

else:

print(i)

**Output:**

1

2

3

4

Hello

6

7

8

9

Hello

11

12

13

14

Hello

16

17

18

19

Hello

21

22

23

24

Hello

26

27

28

29

Hello

31

32

33

34

Hello

36

37

38

39

Hello

41

42

43

44

Hello

46

47

48

49

Q. for i in range(1, 6):

    print(i,"squared is",i\*i)

**Output:**

1 squared is 1

2 squared is 4

3 squared is 9

4 squared is 16

5 squared is 25

Q.Print table of 11 &12.

for i in range(1,11):

    print(11\*i,12\*i)

**Output:**

11 12

22 24

33 36

44 48

55 60

66 72

77 84

88 96

99 108

110 120

Q. class Bike:

    name = ""

    gear = 0

bike1 = Bike()

bike1.gear = 11

bike1.name = "Mountain Bike"

print(f"Name: {bike1.name}, Gears: {bike1.gear}")

**Output:**

Name: Mountain Bike, Gears: 11

Q. class Employee:

    employee\_id = 0

employee1 = Employee()

employee2 = Employee()

employee1.employee\_id = 1001

print(f"Employee ID: {employee1.employee\_id}")

employee2.employee\_id = 1002

print(f"Employee ID: {employee2.employee\_id}")

**Output:**

Employee ID: 1001

Employee ID: 1002

Q.

class Room:

    length = 0.0

    breadth = 0.0

    def calculate\_area(self):

        print("Area of Room =", self.length \* self.breadth)

study\_room = Room()

study\_room.length = 42.5

study\_room.breadth = 30.8

study\_room.calculate\_area()

**Output:**

Area of Room = 1309.0

Q. class University:

cse = 0

cy = 0

aiml = 0

aids = 0

bcom = 0

bba = 0

mba = 0

university1 = University()

university1.cse = 100

university1.cy = 50

university1.aiml = 75

university1.aids = 20

university1.bcom = 120

university1.bba = 90

university1.mba = 80

print("SET")

print(f"CSE: {university1.cse}, CY: {university1.cy}, AIML: {university1.aiml}, AIDs: {university1.aids}")

print("SCM")

print(f"BCom: {university1.bcom}, BBA: {university1.bba}, MBA: {university1.mba}")

**Output:**

SET

CSE: 100, CY: 50, AIML: 75, AIDs: 20

SCM

BCom: 120, BBA: 90, MBA: 80

Q. import math

A = 16

print(math.sqrt(A))

**Output:**

4.0

Q. from math import sqrt, sin

A = 16

B = 3.14

print(sqrt(A))

print(sin(B))

**Output:**

4.0

0.0015926529164868282

Q. import numpy as np

arr = np.array([1, 2, 3])

print("Array with Rank 1: \n", arr)

arr = np.array([[1, 2, 3],

                [4, 5, 6]])

print("Array with Rank 2: \n", arr)

arr = np.array((1, 3, 2))

print("\nArray created using passed tuple:\n", arr)

**Output:**

Array with Rank 1:

[1 2 3]

Array with Rank 2:

[[1 2 3]

[4 5 6]]

Array created using passed tuple:

[1 3 2**]**

Q. import pandas as pd

import numpy as np

ser = pd.Series(dtype='float64')

print("Pandas Series: ", ser)

data = np.array(['g', 'e', 'e', 'k', 's'])

ser = pd.Series(data)

print("Pandas Series:\n", ser)

**Outtput:**

Pandas Series: Series([], dtype: float64)

Pandas Series:

0 g

1 e

2 e

3 k

4 s

dtype: object

Q. from scipy import stats

x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]

y = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]

slope, intercept, r, p, std\_err = stats.linregress(x, y)

print(r)

**Output:**

-0.758591524376155

Q. def add\_numbers():

    a= int(input("Enter first numbers"))

    b= int(input("Enter second numbers"))

    sum=a+b

    print("Sum:",sum)

add\_numbers()

**Output:**

Enter first numbers 5

Enter second numbers 4

Sum: 9

Q.find the sqaure of number using the function

def Square():

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

    b=a\*a

    print("Square=",b)

Square()

**Output:**

Enter first number: 2

Square= 4

Q. write a program in python to print addition of all prime numbers using function

def is\_prime(n):

if n <= 1:

return False

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

if n % i == 0:

return False

return True

def get\_prime\_sum():

total = 0

for i in range(1, 101):

if is\_prime(i):

total += i

print("Sum of all prime numbers from 1 to 100 is:", total)

get\_prime\_sum()

**Output:**

Sum of all prime numbers from 1 to 100 is: 1060

Q. # Input: An integer number

num = int(input("Enter value for fact:"))

# Initialize the factorial variable to 1

factorial = 1

# Calculate the factorial using a for loop

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

 factorial \*= i

# Output: The factorial of the number

print(f"The factorial of {num} is {factorial}")

**Output:**

Enter value for fact:9

The factorial of 9 is 362880

Q. only if condition

a = 30

b = 200

if b > a:

 print("b is greater than a")

**Output:**

b is greater than a

Q. if-else condition

i = 20

if (i > 0):

 print("i is positive")

else:

 print("i is 0 or Negative")

**Output:**

i is positive

Q. slicing operator

my\_list = ['p', 'r', 'o', 'g', 'r', 'a', 'm']

print("my\_list =", my\_list)

print("my\_list[2: 5] =", my\_list[2: 5])

print("my\_list[2: -2] =", my\_list[2: -2])

print("my\_list[0: 3] =", my\_list[0: 3])

**Output:**

my\_list = ['p', 'r', 'o', 'g', 'r', 'a', 'm']

my\_list[2: 5] = ['o', 'g', 'r']

my\_list[2: -2] = ['o', 'g', 'r']

my\_list[0: 3] = ['p', 'r', 'o']

Q.list example

fruits = ['apple', 'grapes', 'oranges']

print("Original List:", fruits)

fruits.insert(2, 'cherry')

print("After Insert:", fruits)

fruits.append('banana')

print("After Append:", fruits)

fruits.remove('apple')

print("After Remove:", fruits)

fruits.clear()

print("After Clear:", fruits)

fruits.extend('kiwi')

print("After Extend:", fruits)

fruits.sort()

print("After Sort:", fruits)

fruits.reverse()

print("After Reverse:", fruits)

copied\_fruits = fruits.copy()

print("Copied List:", copied\_fruits)

fruits.pop()

print("After Pop:", fruits)

**Output:**

Original List: ['apple', 'grapes', 'oranges']

After Insert: ['apple', 'grapes', 'cherry', 'oranges']

After Append: ['apple', 'grapes', 'cherry', 'oranges', 'banana']

After Remove: ['grapes', 'cherry', 'oranges', 'banana']

After Clear: []

After Extend: ['k', 'i', 'w', 'i']

After Sort: ['i', 'i', 'k', 'w']

After Reverse: ['w', 'k', 'i', 'i']

Copied List: ['w', 'k', 'i', 'i']

After Pop: ['w', 'k', 'i']

Q.

country\_capitals = {"United States": "Washington D.C.","Italy": "Rome"}

for country in country\_capitals:

 print(country)

print()

for country in country\_capitals:

 capital = country\_capitals[country]

 print(capital)

**Output:**

United States

Italy

Washington D.C.

Rome

Q. Calculator

def add(x, y):

    return x + y

def subtract(x, y):

    return x - y

def multiply(x, y):

    return x \* y

def divide(x, y):

    return x / y

print("Select operation.")

print("1. Add")

print("2. Subtract")

print("3. Multiply")

print("4. Divide")

while True:

    choice = input("Enter choice(1/2/3/4): ")

    if choice in ('1', '2', '3', '4'):

        try:

            num1 = float(input("Enter first number: "))

            num2 = float(input("Enter second number: "))

        except ValueError:

            print("Invalid input. Please enter a number.")

            continue

        if choice == '1':

            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':

            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':

            print(num1, "\*", num2, "=", multiply(num1, num2))

        elif choice == '4':

            if num2 != 0:

                print(num1, "/", num2, "=", divide(num1, num2))

            else:

                print("Error! Division by zero is not allowed.")

        next\_calculation = input("Let's do next calculation? (yes/no): ")

        if next\_calculation.lower() == "no":

            break

    else:

        print("Invalid Input")

**Output:**

Select operation.

1. Add

2. Subtract

3. Multiply

4. Divide

Enter choice(1/2/3/4): 2

Enter first number: 6

Enter second number: 8

6.0 - 8.0 = -2.0

Let's do next calculation? (yes/no): y

Q. Take two list and form the third list by using merge operation.

fruits = ["apple", "mango", "cherry"]

flowers = ["rose", "lily", "lotus"]

merged\_list = fruits + flowers

print("Merged List:", merged\_list)

Output:

Merged List: ['apple', 'mango', 'cherry', 'rose', 'lily', 'lotus']

Q. Print sq.no upto 10 and there sum.

newlist = [x\*\*2 for x in range(1, 11) if x\*\*2 <= 10]

sum\_of\_squares = sum(newlist)

print(newlist)

print(sum\_of\_squares)

**Output:**

[1, 4, 9]

14

Q. Find sq of any no.using function.

def square\_number(num):

return num \* num

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

print("Square:", square\_number(num))

**Output:**

Enter a number: 4

Square: 16

Que) prepare a code in which fruit a is present.

fruits = ["apple", "mango", "cherry", "kiw]

print("Fruits containing 'a':")

for fruit in fruits:

    if "a" in fruit:

        print(fruit)

**Output:**

Fruits containing 'a':

apple

mango

Que) display items from stack 1 to -4 and -5 to 0

stack = [-1, 1 ,3, -8, 7 ]

part1 = stack[1:-4]

print("Items from index 1 to -4:", part1)

part2 = stack[-5:1]

print("Items from index -5 to 0:", part2)

**Output:**

Items from index 1 to -4: []

Items from index -5 to 0: [-1]

Q. for x in range(100,0,-1):

    print (x)

print("Blastoff!")

Q. import matplotlib.pyplot as plt

 from scipy import stats

 x = [5,7,8,7,2,17,2,9,4,11,12,9,6]

 y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

 slope, intercept, r, p, std\_err = stats.linregress(x, y)

 def myfunc(x):

  return slope \* x + intercept

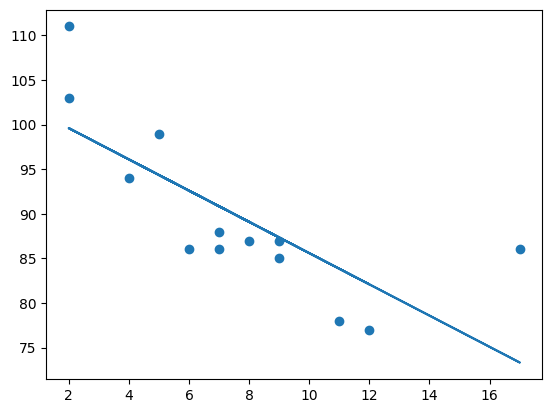
 mymodel = list(map(myfunc, x))

 plt.scatter(x, y)

 plt.plot(x, mymodel)

 plt.show()

**Output:**



Q. f = open(r"C:\Users\Shrawani\Desktop\file.txt.txt","r")

print(f.read())

**Output:**

Hello everyone!

I am student of Sanjivani University.

Today I am going to learn the new concept named as "File Handling"

Q. f = open(r"C:\Users\Shrawani\Desktop\file.txt.txt","w")

f.write("Hello Mansi!")

f.close()

f = open(r"C:\Users\Shrawani\Desktop\file.txt.txt","r")

print(f.read())

f.close()

**Output:**

Hello Mansi!