1. # Python program for sum of the square of first N natural numbers

# Getting input from users

N = int(input("Enter value of N: "))

# calculating sum of square

sumVal = 0

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

sumVal += (i\*i)

print("Sum of squares = ", sumVal)

**2. merge two files**

print("Enter the Name of First File: ", end="")

fileOne = input()

print("Enter the Name of Second File: ", end="")

fileTwo = input()

print("Enter the Name of Third File: ", end="")

fileThree = input()

content = " "

fh = open(fileOne, "r")

for line in fh:

content = content + line + '\n'

fh.close()

fh = open(fileTwo, "r")

for line in fh:

content = content + line + '\n' fh.close()

fh = open(fileThree, "w")

fh.write(content)

print("\nFile merged successfully!")

**3. To remove punctuation**

# define punctuation

punctuations = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

my\_str = "Hello!!!, he said ---and went."

# To take input from the user

# my\_str = input("Enter a string: ")

# remove punctuation from the string

no\_punct = ""

for char in my\_str:

if char not in punctuations:

no\_punct = no\_punct + char

# display the unpunctuated string

print(no\_punct)

**4. Tkinter Module**

import tkinter as tk

def write\_text():

print("Tkinter is easy to create GUI!")

parent = tk.Tk()

frame = tk.Frame(parent)

frame.pack()

text\_disp= tk.Button(frame,

text="Hello",

command=write\_text)

text\_disp.pack(side=tk.LEFT)

exit\_button = tk.Button(frame,

text="Exit",

fg="green",

command=quit)

exit\_button.pack(side=tk.RIGHT)

parent.mainloop()

**5. Mathematical expression**

**import** math

# initializing the value

test\_int **=** 4

test\_neg\_int **=** **-**3

test\_float **=** 0.00

# checking exp() values

# with different numbers

**print** (math.exp(test\_int))

**print** ("math.floor(-23.11) : ", math.floor(**-**23.11))

print(math.factorial(23))

print(math.factorial(23))

print(pow(3, 4, 10))

**print** (math.cos(a))

**6. Operations on tuples**

# Different types of tuples

# Empty tuple

my\_tuple = ()

print(my\_tuple)

# Tuple having integers

my\_tuple = (1, 2, 3)

print(my\_tuple)

# tuple with mixed datatypes

my\_tuple = (1, "Hello", 3.4)

print(my\_tuple)

# nested tuple

my\_tuple = ("mouse", [8, 4, 6], (1, 2, 3))

print(my\_tuple)

# 7. Python Program to check Armstrong Number

# Python program to determine whether

# the number is Armstrong number or not

# Function to calculate x raised to

# the power y

def power(x, y):

if y == 0:

return 1

if y % 2 == 0:

return power(x, y // 2) \* power(x, y // 2)

return x \* power(x, y // 2) \* power(x, y // 2)

# Function to calculate order of the number

def order(x):

# Variable to store of the number

n = 0

while (x != 0):

n = n + 1

x = x // 10

return n

# Function to check whether the given

# number is Armstrong number or not

def isArmstrong(x):

n = order(x)

temp = x

sum1 = 0

while (temp != 0):

r = temp % 10

sum1 = sum1 + power(r, n)

temp = temp // 10

# If condition satisfies

return (sum1 == x)

# Driver code

x = 153

print(isArmstrong(x))

x = 1253

print(isArmstrong(x))

8.Wite a program to create histogram using matplotlib

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.random.normal(170, 10, 250)  
  
plt.hist(x)  
plt.show()

9. write a program to calculate simple and compound interest in python

# Reading principal amount, rate and time

principal = float(input('Enter amount: '))

time = float(input('Enter time: '))

rate = float(input('Enter rate: '))

# Calcualtion

simple\_interest = (principal\*time\*rate)/100

compound\_interest = principal \* ( (1+rate/100)\*\*time - 1)

# Displaying result

**print**('Simple interest is: %f' % (simple\_interest))

**print**('Compound interest is: %f' %(compound\_interest))

10. Write a program to demonstrate the operation on list in python

# Python program to demonstrate

# Creation of List

# Creating a List

List = []

print("Blank List: ")

print(List)

# Creating a List of numbers

List = [10, 20, 14]

print("\nList of numbers: ")

print(List)

# Creating a List of strings and accessing

# using index

List = ["Geeks", "For", "Geeks"]

print("\nList Items: ")

print(List[0])

print(List[2])

# Python program to demonstrate

# accessing of element from list

# Creating a List with

# the use of multiple values

List = ["Geeks", "For", "Geeks"]

# accessing a element from the

# list using index number

print("Accessing a element from the list")

print(List[0])

print(List[2])

# Python program to demonstrate

# Addition of elements in a List

# Creating a List

List = []

print("Initial blank List: ")

print(List)

# Addition of Elements

# in the List

List.append(1)

List.append(2)

List.append(4)

print("\nList after Addition of Three elements: ")

print(List)

# Adding elements to the List

# using Iterator

for i in range(1, 4):

List.append(i)

print("\nList after Addition of elements from 1-3: ")

print(List)

# Adding Tuples to the List

List.append((5, 6))

print("\nList after Addition of a Tuple: ")

print(List)

# Addition of List to a List

List2 = ['For', 'Geeks']

List.append(List2)

print("\nList after Addition of a List: ")

print(List)

# Python program to demonstrate

# Addition of elements in a List

# Creating a List

List = [1,2,3,4]

print("Initial List: ")

print(List)

# Addition of Element at

# specific Position

# (using Insert Method)

List.insert(3, 12)

List.insert(0, 'Geeks')

print("\nList after performing Insert Operation: ")

print(List)

11.write a program to demonstrate the use of recursive function

def factorial(x):

"""This is a recursive function

to find the factorial of an integer"""

if x == 1:

return 1

else:

return (x \* factorial(x-1))

num = 3

print("The factorial of", num, "is", factorial(num))

12.Write a program to connect to database and perform insert and delete operation.

import sqlite3

# connecting to the database

connection = sqlite3.connect("gfg.db")

# cursor

crsr = connection.cursor()

# print statement will execute if there

# are no errors

print("Connected to the database")

# close the connection

connection.close()

import sqlite3

# connecting to the database

connection = sqlite3.connect("gfg.db")

# cursor

crsr = connection.cursor()

# SQL command to create a table in the database

sql\_command = """CREATE TABLE emp (

staff\_number INTEGER PRIMARY KEY,

fname VARCHAR(20),

lname VARCHAR(30),

gender CHAR(1),

joining DATE);"""

# execute the statement

crsr.execute(sql\_command)

# close the connection

connection.close()

# Python code to demonstrate table creation and

# insertions with SQL

# importing module

import sqlite3

# connecting to the database

connection = sqlite3.connect("gfg.db")

# cursor

crsr = connection.cursor()

# SQL command to insert the data in the table

sql\_command = """INSERT INTO emp VALUES (23, "Rishabh",\

"Bansal", "M", "2014-03-28");"""

crsr.execute(sql\_command)

# another SQL command to insert the data in the table

sql\_command = """INSERT INTO emp VALUES (1, "Bill", "Gates",\

"M", "1980-10-28");"""

crsr.execute(sql\_command)

# To save the changes in the files. Never skip this.

# If we skip this, nothing will be saved in the database.

connection.commit()

# close the connection

connection.close()

# importing module

import sqlite3

# connecting to the database

connection = sqlite3.connect("gfg.db")

# cursor

crsr = connection.cursor()

# primary key

pk = [2, 3, 4, 5, 6]

# Enter 5 students first names

f\_name = ['Nikhil', 'Nisha', 'Abhinav', 'Raju', 'Anshul']

# Enter 5 students last names

l\_name = ['Aggarwal', 'Rawat', 'Tomar', 'Kumar', 'Aggarwal']

# Enter their gender respectively

gender = ['M', 'F', 'M', 'M', 'F']

# Enter their joining data respectively

date = ['2019-08-24', '2020-01-01', '2018-05-14', '2015-02-02', '2018-05-14']

for i in range(5):

# This is the q-mark style:

crsr.execute('INSERT INTO emp VALUES ({pk[i]}, "{f\_name[i]}", "{l\_name[i]}", "{gender[i]}", "{date[i]}")')

# To save the changes in the files. Never skip this.

# If we skip this, nothing will be saved in the database.

connection.commit()

# close the connection

connection.close()

# Import module

import sqlite3

# Connecting to sqlite

conn = sqlite3.connect('gfg.db')

# Creating a cursor object using

# the cursor() method

cursor = conn.cursor()

# Updating

cursor.execute('''DELETE FROM emp WHERE fname="Rishabh";''')

# Commit your changes in the database

conn.commit()

# Closing the connection

conn.close()0