**NAME-SAQIB AHMAD**

**COURSE-MCA-3A**

**STUDENT ID-20711061**

**Assignment**

**Q1: - Write a program to use the mathematical operators.**

**Code:**

num1=int(input("Enter the first number:"))

num2=int(input("Enter the second number:"))

print(num1 + num2)

print(num1 - num2)

print(num1 / num2)

print(num1 \* num2)

print(num1 % num2)

print(num1 \*\* num2)

print(num1 // num2)

**Output:**

Graphical user interface, text, application

Description automatically generated

**Q2: - Write a program to take an input of numbers from the user and print the Fibonacci series to the terminal number.**

**Code:**

nterms = int(input("Number of terms"))

n1, n2 = 0, 1

count = 0

if nterms <= 0:

print("Please enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence upto",nterms,":")

print(n1)

else:

print("Fibonacci sequence:")

while count < nterms:

print(n1)

nth = n1 + n2

n1 = n2

n2 = nth

count += 1

**Output:**

Text

Description automatically generated

**Q3: - Write a program to print the factorial of the number input by the user.**

**Code:**

num = int(input("enter the number : "))

factorial = 1

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

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

factorial = factorial\*i

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

**Output:**

Graphical user interface

Description automatically generated

**Q4: - Write a program to check whether a given number is a prime number or not using loops.**

**Code:**

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

flag = False

if num > 1:

for i in range(2, num):

if (num % i) == 0:

flag = True

break

if flag:

print(num, "is not a prime number")

else:

print(num, "is a prime number")

**Output:**

Graphical user interface, application

Description automatically generated

**Q5: - Write a program to demonstrate the importing of modules of python.**

**Code:**

from math import pi

print(pi)

**Output:**



**Q6: - Write a program to demonstrate the use of nested if statements.**

**Code:**

var = 100

if var < 200:

print ("Expression value is less than 200")

if var == 150:

print ("Which is 150")

elif var == 100:

print ("Which is 100")

elif var == 50:

print ("Which is 50")

elif var < 50:

print ("Expression value is less than 50")

else:

print ("Could not find true expression")

**Output:**

Graphical user interface, text

Description automatically generated with medium confidence

**Q7: - Write a program to demonstrate the use of the else clause.**

**Code:**

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

if(num>0):

print("positive number")

else:

print("negative number")

**Output:**

Graphical user interface

Description automatically generated with medium confidence

**Q8: - Write a program to illustrate the usage of Tuples.**

**Code:**

my\_tuple = ()

print(my\_tuple)

my\_tuple = (1, 2, 3)

print(my\_tuple)

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

print(my\_tuple)

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

print(my\_tuple)

**Output:**

Text

Description automatically generated

**Q9: - Write a program for searching an element and sorting a List.**

**Code:**

list = [7,8,90,86,56]

list.sort()

print(list)

n = int(input("Enter the element to be searched : "))

print("Element found at index : ",list.index(n))

**Output:**

Text

Description automatically generated

**Q10: - Write a program to illustrate the usage of Dictionaries.**

**Code:**

Dict = {1: 'Saqib', 2: 'Sagar', 3: 'Ankit'}

print("\nDictionary with the use of Integer Keys: ")

print(Dict)

Dict = {'Name': 'Saqib', 1: [1, 2, 3, 4]}

print("\nDictionary with the use of Mixed Keys: ")

print(Dict)

**Output:**

Text

Description automatically generated

**Q11: - Write a program to find the mean. mode and median of the given range of numbers.**

**Code:**

import numpy

from scipy import stats

age = [22,23,24,21,20,18,19,23,22,16,17,20]

print("mean = ",numpy.mean(age))

print("median = ",numpy.median(age))

print("mode = ",stats.mode(age))

**Output:**

Text

Description automatically generated with medium confidence

**Q12: - Write a program to calculate the standard deviation of a given set of numbers.**

**Code:**

import numpy

age = [22,23,24,21,20,18,19,23,22,16,17,20]

print("Standard deviation is :",numpy.std(age))

**Output:**

Graphical user interface, application

Description automatically generated

**Q13: - Write a program to calculate the addition of two 3x 3 matrices.**

**Code:**

X = [[1,1,3],

[4 ,5,9],

[7 ,0,7]]

Y = [[9,2,7],

[8,0,4],

[9,2,1]]

result = [[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(X)):

for j in range(len(X[0])):

result[i][j] = X[i][j] + Y[i][j]

for r in result:

print(r)

**Output:**

Graphical user interface, text, application

Description automatically generated

**Q14: - Write a program to calculate the multiplication of two 3x 3 matrices.**

**Code:**

X = [[1,1,3],

[4 ,5,9],

[7 ,0,7]]

Y = [[9,2,7],

[8,0,4],

[9,2,1]]

result = [[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(X)):

for j in range(len(Y[0])):

for k in range(len(Y)):

result[i][j] += X[i][k] \* Y[k][j]

for r in result:

print(r)

**Output:**

A screenshot of a computer

Description automatically generated with low confidence

**Q15: - Write a program to calculate the inverse of the given matrix.**

**Code:**

import numpy

x=numpy.array([[5,9], [16,7]])

y=numpy.linalg.inv(x)

print(x)

print (y)

**Output:**

Graphical user interface, text, application

Description automatically generated

**Q16: - Write a program to calculate the transpose of the given matrix.**

**Code:**

X = [[5,9],

[7 ,16],

[12 ,28]]

result = [[0,0,0],

[0,0,0]]

for i in range(len(X)):

for j in range(len(X[0])):

result[j][i] = X[i][j]

for r in result:

print(r)

**Output:**

Graphical user interface, text, application

Description automatically generated