**LAB WORK**

**Of**

**MACHINE LEARNING**

**USING PYTHON**

**PMC 303**

****

**Submitted By:**

**Name- Sana Parveen**

**Student Id- 20712141**

**Course- MCA**

**Semester – 3**

**Section- C**

**INDEX**

|  |  |
| --- | --- |
| **S. No.** | **Programs** |
| 1. | WAP to use mathematical operators. |
| 2. | WAP to take an input from user and print fibonacci series upto terminal number. |
| 3. | WAP to print factorial of a number input by the user. |
| 4. | WAP to check whether a number is prime or not using loops. |
| 5. | WAP to demonstrate the use of nested if statements. |
| 6. | WAP to demonstrate the use of nested else statements. |
| 7. | WAP for searching an element and sorting a list. |
| 8. | WAP to find mean,median and mode of the given numbers. |
| 9. | WAP to calculate standard deviation of given numbers. |
| 10. | WAP to calculate addition of two 3\*3 matrices. |
| 11. | WAP to calculate multiplication of two 3\*3 matrices. |
| 12. | WAP to calculate inverse of a matrix. |
| 13. | WAP to calculate transpose of a matrix. |

Q1. WAP to use mathematical operators.

=>

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

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

sum = num1 + num2

print("Sum is: ",sum)

difference = num1 - num2

print("Difference is: ",difference)

product = num1 \* num2

print("Product is: ",product)

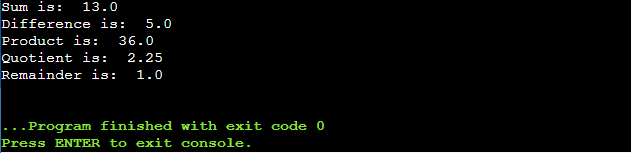
quotient = num1 / num2

print("Quotient is: ",quotient)

remainder = num1 % num2

print("Remainder is: ",remainder)

**OUTPUT-**



Q2. WAP to take a input from user and print fibonacci series to the terminal number.

=>

n = int(input("Enter the number of terms: "))

n1,n2 = 0,1

count = 0

if n <= 0:

print("Invalid Input")

elif n == 1:

print(n1)

else:

print("Fibonacci sequence:")

while count <n:

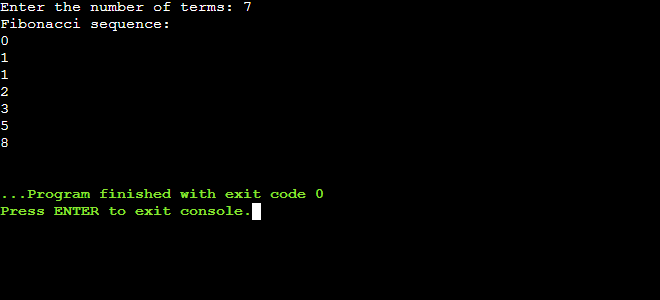
print(n1)

n3 = n1 + n2

n1 = n2

n2 = n3

count += 1

**OUTPUT-**

Q3. WAP to print factorial of a number.

=>

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

factorial = 1

if num < 1 :

print("Invalid input")

elif num == 1:

print("The factorial of 0 is 1")

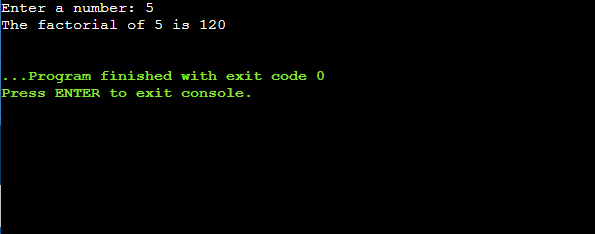
else:

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

factorial = factorial \* i

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

**OUTPUT-**



Q4. WAP to check whether a number is prime or not using loops.

=>

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

count = 0

for i in range(2,(num//2 + 1)):

if(num%i == 0):

count += 1

break

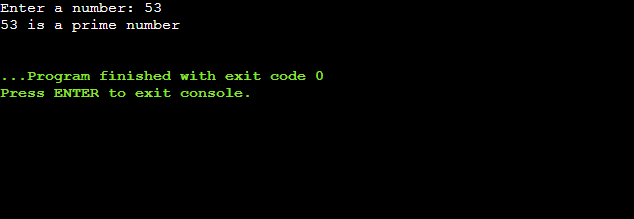
if(count == 0 and num != 0):

print(num,"is a prime number")

else:

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

**OUTPUT-**



Q5. Write a program to demonstrate the use of nested if statements.

=>

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

if num >= 0:

if num == 0:

print("Zero")

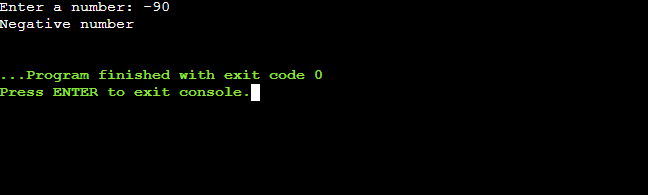
else:

print("Positive number")

else:

print("Negative number")

**OUTPUT-**



Q6. Write a program to demonstrate use of nested else.

=>

num = 20

if(num==20):

print("Number is 10")

elif(num==15):

print("Number is 15")

elif(num==20):

print("Number is 20")

else:

print("Number is not present")

**OUTPUT-**



Q7.WAP for searching an element and sorting a list.

=>

numbers = [2,6,5,4,9,8,2,1,3,7]

numbers.sort()

print(numbers)

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

for i in numbers:

if i==n:

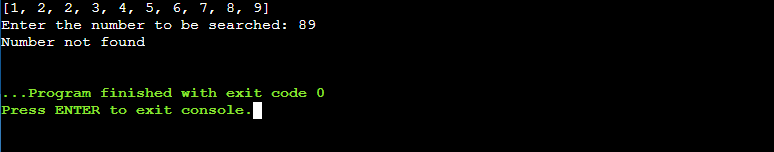
print("Number found")

break

else:

print("Number not found")

**OUTPUT-**



Q8.WAP to find the mean,median and mode of the given range of numbers.

=>

from collections import Counter

list1 = [5,9,6,4,8,6,7,2,1,3]

n = len(list1)

sum = sum(list1)

mean = sum/n

print("Mean is: "+str(mean))

list1.sort()

if n%2 == 0:

median1 = list1[n//2]

median2 = list1[n//2 - 1]

median = (median1 + median2)/2

else:

median = list1[n//2]

print("Median is: "+str(median))

data = Counter(list1)

get\_mode = dict(data)

mode = [k for k,v in get\_mode.items() if v== max(list(data.values()))]

if len(mode) == n:

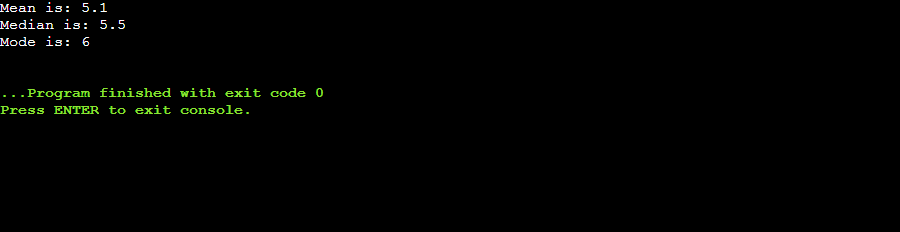
get\_mode = "No mode found"

else:

get\_mode ="Mode is: "+','.join(map(str,mode))

print(get\_mode)

**OUTPUT-**



Q9. WAP to find the standard deviation of given numbers.

=>

numbers =[1,5,4,2,0]

sum = 0

for i in range(len(numbers)):

sum += numbers[i]

mean = sum/len(numbers)

sum\_of\_squared\_deviation = 0

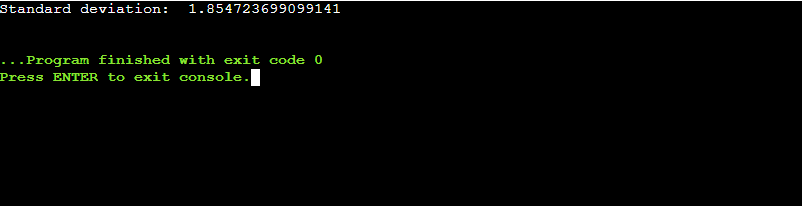
for i in range(len(numbers)):

sum\_of\_squared\_deviation += (numbers[i]- mean)\*\*2

Standard\_deviation = ((sum\_of\_squared\_deviation)/len(numbers))\*\*0.5

print("Standard deviation: ",Standard\_deviation)

**OUTPUT-**



Q10. WAP to calculate addition of two 3\*3 matrix.

=>

x = [[1,2,3],

[4,5,6],

[7,8,9]]

y = [[9,8,7],

[6,5,4],

[3,2,1]]

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-**



Q11. WAP to calculate multiplication of two 3\*3 matrix.

=>

x = [[1,2,3],

[4,5,6],

[7,8,9]]

y = [[9,8,7],

[6,5,4],

[3,2,1]]

result = [[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(x)):

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

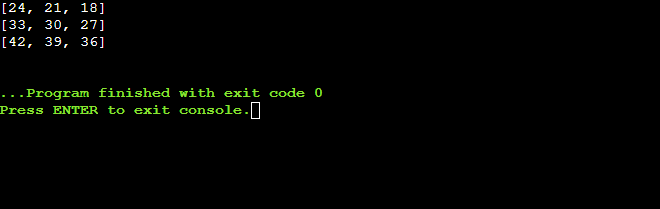
for k in range(len(y)):

result[i][j] += x[i][k] + y[k][j]

for r in result:

print(r)

**OUTPUT-**



Q12. WAP to find inverse of a matrix .

=>

import numpy as np

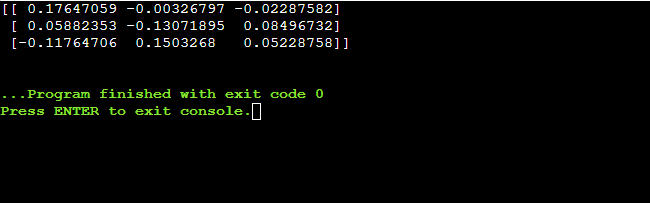
x =np.array([[6,1,1],

[4,-2,5],

[2,8,7]])

print(np.linalg.inv(x))

**OUTPUT-**



Q13. WAP to find transpose of a matrix.

=>

x=[[12,7],

[4,5],

[3,8]]

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-**

