Dr.M.G.R.
Educational and Research Institute
(DEEMED TO BE UNIVERSITY)
(An ISO Certified Institution)
University with Graded Autonomy Status
Maduravoyal , Chennai - 600 095

LAB MANUAL

EBCS22ET2 – PYTHON PROGRAMMING

2024-2025 (EVEN SEMESTER)

DEPARTMENT

OF

ENGINEERING AND TECHNOLOGY

NAME   :

REG. NO.   :

COURSE   :

YEAR/SEM   :

List of Experiments

1. Develop a python program to find the area and circumference of a circle.

2. Develop a python program to check if the number is positive or negative or zero using nested if else statement.

3. Develop a python program to find the GCD (Greatest Common Divisor) of two numbers.

4. Develop a Python program using function to compute the factorial of a given number.

5. Develop a Python program to find the sum of square of individual digits of a number using function.

6. Develop a Python program to find the largest digit from a number using function.

7. Develop a Python program to display only the positive elements of the list.

8. Develop a Python program to accept any number and print it in words.

9. Develop a Python program to subtract two matrices.

10. Develop a Python program to perform matrix multiplication.

INDEX

Date :

Exp.no: 01

Develop a python program to find the area and circumference of a circle.

Aim :

      To Develop a python program to find the area and circumference of a circle.

Algorithm :

    STEP 1: Allow users to input the radius of a circle.

    STEP 2: Using the radius value, this Python formula to calculate the Circumference and Area of a Circle.

    STEP 3: Area of a circle are: $A = \pi r^2 = \pi *$ radius $*$ radius.

    STEP 4: Circumference of a Circle $= 2\pi r = 2 * \pi *$ radius.

    STEP 5: Print result.

Flow Chart :

```
                      ┌─────────────┐
                      │    START    │
                      └─────────────┘
                             │
                             ▼
                    ╱─────────────────╱
                   ╱   ENTER RADIUS  ╱
                  ╱─────────────────╱
                             │
                             ▼
                  ┌──────────────────────────┐
                  │ Area = PI* radius *radius │
                  │                          │
                  │ Circumference = 2* PI*radius │
                  └──────────────────────────┘
                             │
                             ▼
                    ╱─────────────────╱
                   ╱  Display area and ╱
                  ╱   circumference   ╱
                 ╱─────────────────╱
                             │
                             ▼
                      ┌─────────────┐
                      │     END     │
                      └─────────────┘
```

2
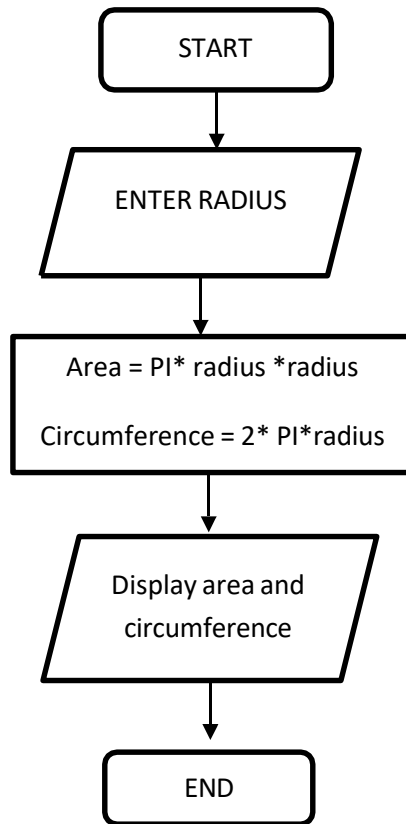
Code :

```
PI = 3.14
radius = float(input(' Please Enter the radius of a circle: '))
circumference = 2 * PI * radius
area = PI * radius * radius
print(" Circumference Of a Circle = %.2f" %circumference)
print(" Area Of a Circle = %.2f" %area)
```

Output :

Result :

Date :

Exp.no: 02

Develop a python program to check if the number is positive or negative or zero using nested if else statement

Aim :

     To develop a python program to check if the number is positive or negative or zero using nested if else statement.

Algorithm :

    STEP 1: Allow users to input a number specifying the float data type.

    STEP 2: Use nested if else statement, initially start a if condition that checks if the num is greater than 0, if true display that the number is positive.

    STEP 3: Use elif that checks if the num is 0, if true display that it is zero.

    STEP 4: When both the condition is false then else statement will be executed as the number is negative.

Flow Chart:
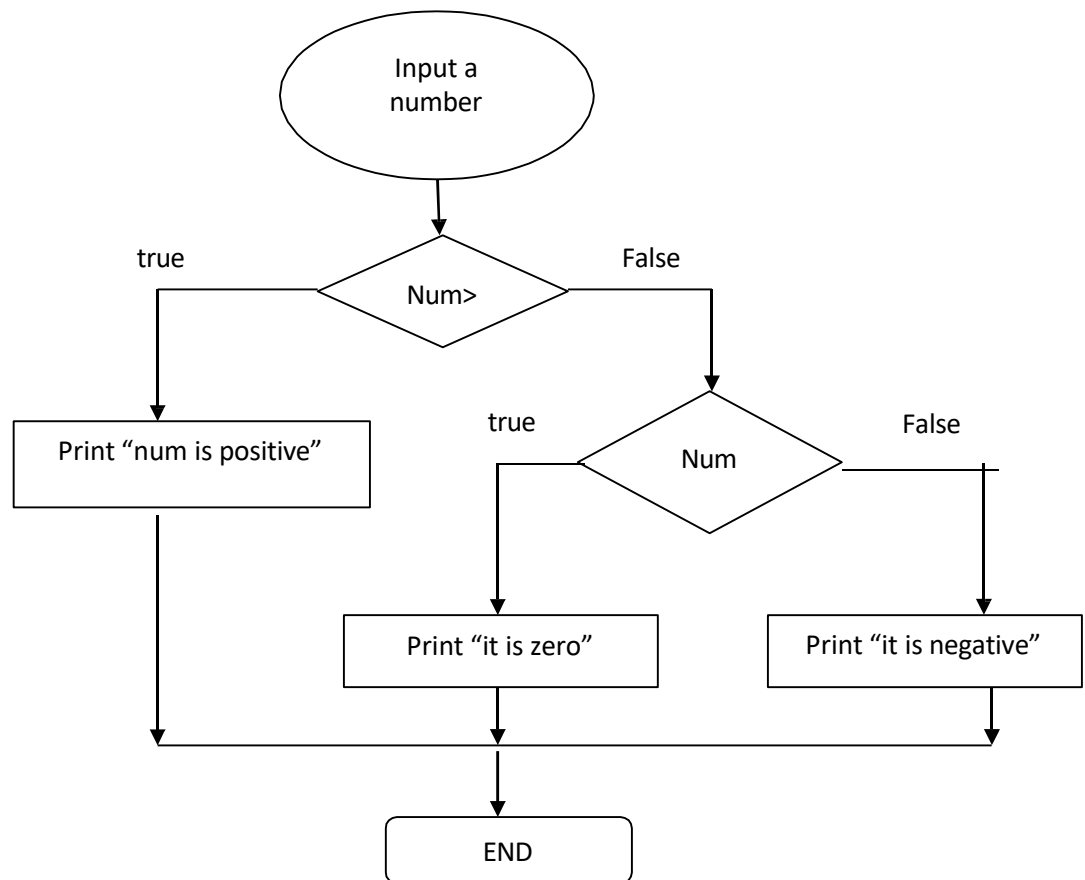
Code :

```
Code : num = float(input("Input a number: "))
if num > 0:
        print("It is positive number")
elif num == 0:
        print("It is Zero")
else:
        print("It is a negative number")
```

Output :

Result :

8

Date :

Exp.no: 03

Develop a python program to find the GCD (Greatest Common Divisor) of two numbers.

Aim :

To develop a python program to find the GCD (Greatest Common Divisor) of two numbers.

Algorithm :

STEP 1: Start the program with importing math.

STEP 2: Get the input for num1 and num2.

STEP 3: Use math.gcd to find the gcd of num1 and num2
gcd=math.gcd(num1,num2)

STEP 4: Print the gcd.

Flow Chart :

```
            ┌─────────────┐
            │   START     │
            └──────┬──────┘
                   │
                   ▼
           ╱─────────────╱
          ╱  Enter two  ╱
         ╱   numbers   ╱
        ╱─────────────╱
                   │
                   ▼
     ┌───────────────────────────┐
     │  Use math.gcd to find gcd │
     │                           │
     │ gcd=math.gcd(num1, num2)  │
     └─────────────┬─────────────┘
                   │
                   ▼
          ╱─────────────╱
         ╱ Display gcd ╱
        ╱─────────────╱
                   │
                   ▼
            ┌─────────────┐
            │    END      │
            └─────────────┘
```
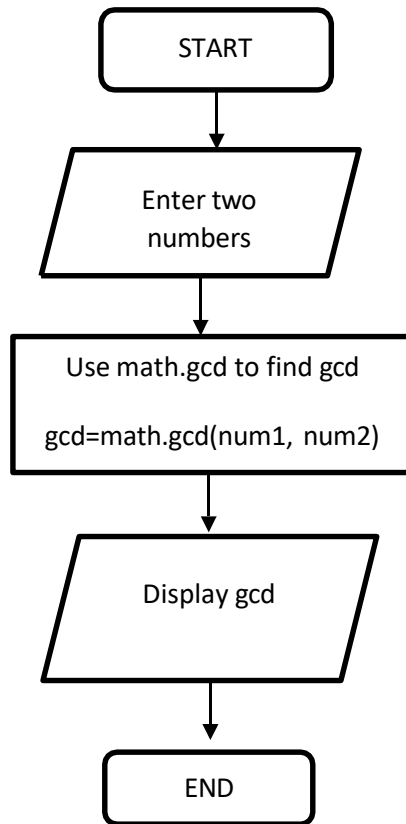
Code :

```
import math

num1=int(input("num1=")

num2=int(input("num2=")

gcd=math.gcd(num1,num2)

print("GCD of ",num1,"and",num2, "is",gcd)
```

Output :

Result :

Date :

Exp.no: 04

Develop a Python program using function to compute the factorial of a given number.

Aim :

To develop a Python program using function to compute the factorial of a given number.

Algorithm :

STEP 1: Start

STEP 2: Read the input number from the user

STEP 3: Declare and initialize variables fact = 1 and i = 1

STEP 4: Repeat the loop until i<=num

fact = fact * i

i = i++

STEP 5: Print fact to get the factorial of a given number

STEP 6: Stop

Flow Chart :

Start

Input N

Fact=1

For i = 1 to N

Fact= Fact * i

i

Print Fact

Stop
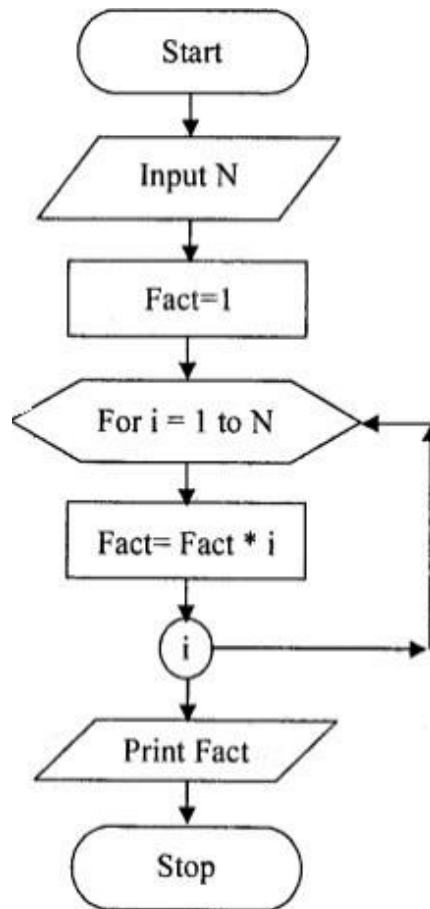
Code :

```
n = int(input("Enter a number: "))
fact = 1
for i in range(1,n + 1):
    fact = fact*i
print("The factorial of",n,"is",fact)
```

Output :

Result :

Date :

Exp.no: 05

Develop a Python program to find the sum of square of individual digits of a number using function

Aim :

To develop a Python program to find the sum of square of individual digits of a number using function

Algorithm :

STEP 1: Initialize sum_of_squares to 0.

STEP 2: Start a while loop with the condition number > 0.

STEP 3: Within the while loop, find the last digit of the number by taking the modulo 10 of the number and assign it to the variable digit.

STEP 4: Calculate the square of the digit using the expression digit ** 2 and add it to the sum_of_squares.

STEP 5: Update the number by integer division with 10.

STEP 6: Continue the loop until number is no longer greater than 0.

STEP 7: Return the value of sum_of_squares.

Flow Chart :

```
              ┌─────────────┐
              │    START    │
              └──────┬──────┘
                     │
                     ▼
            ╱─────────────────╲
           ╱   Enter an input  ╱
          ╱──────────┬────────╱
                     │
                     ▼
         ┌───────────────────────────┐
         │ Initialize sum_of_squares │
         │           to 0            │
         └─────────────┬─────────────┘
                       │
                       ▼
                   ◇ While ◇
                   ◇Number >◇ ──────────┐
                       │                │
                       ▼                │
         ┌───────────────────────────┐  │
         │    Digit = number%10      │  │
         └─────────────┬─────────────┘  │
                       │                │
                       ▼                │
         ┌───────────────────────────┐  │
         │ Sum_of_squares += digit **2│ │
         └─────────────┬─────────────┘  │
                       │                │
                       ▼                │
         ┌───────────────────────────┐  │
         │       Number //= 10       │  │
         └─────────────┬─────────────┘  │
                       │◄───────────────┘
                       ▼
              ┌─────────────┐
              │     END     │
              └─────────────┘
```
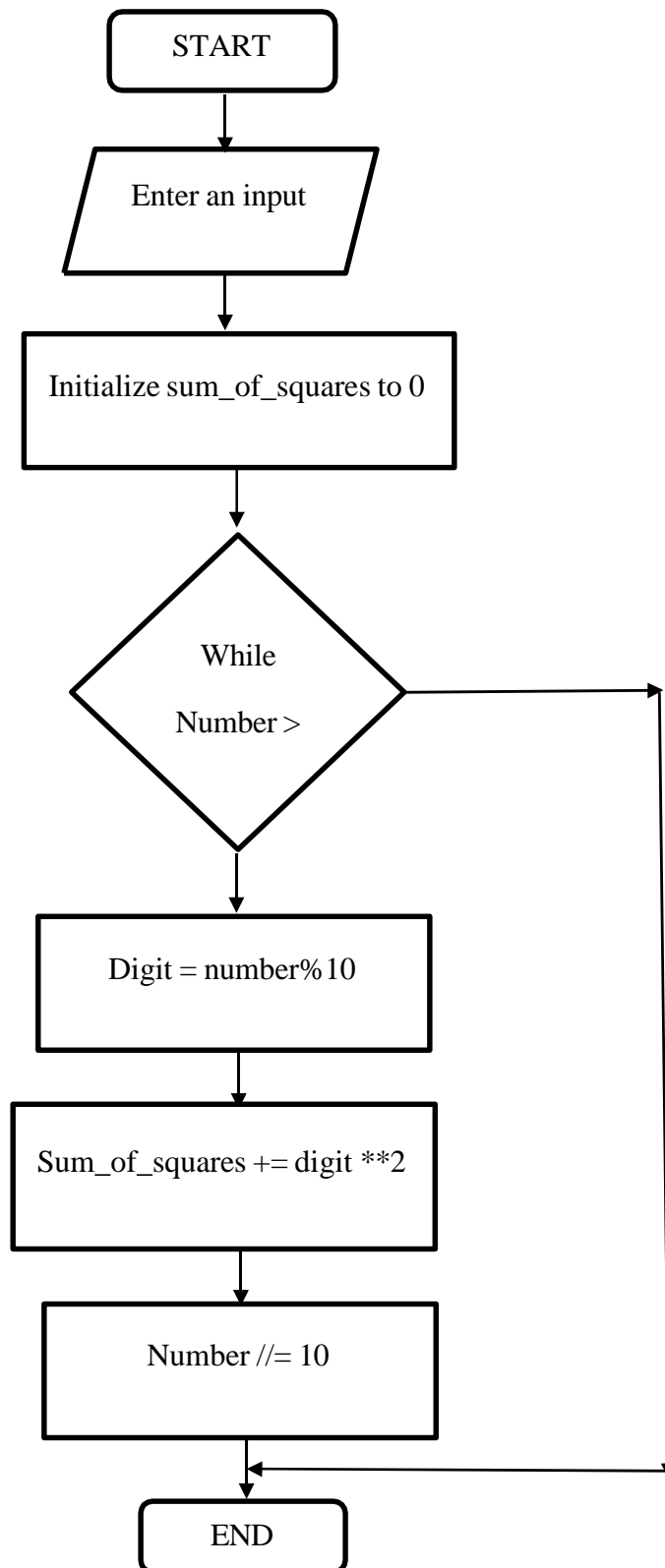
18

Code :

```
def sum_of_digit_squares(number):

    sum_of_squares = 0

    while number > 0:

        digit = number % 10

        sum_of_squares += digit ** 2

        number //= 10

    return sum_of_squares

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


print("The sum of the squares of the digits in", number, "is",
sum_of_digit_squares(number))
```

Output :

Result :

Output :

Date :

Exp.no: 06

Develop a Python program to find the largest digit from a number using function.

Aim :

   To develop a Python program to find the largest digit from a number using function

Algorithm :

   STEP 1: Start

   STEP 2: Set a variable max_digit to 0

   STEP 3: While n is greater than 0, do the following:

      a. Get the last digit of n using the modulus operator (%) and assign it to a variable called digit

      b. If digit is greater than max_digit, set max_digit equal to digit

      c. Divide n by 10 using integer division (//) to remove the last digit

   STEP 4: Return max_digit

   STEP 5: Call the function largest_digit with num as an argument and print the result

   STEP 6: End

Flow Chart :
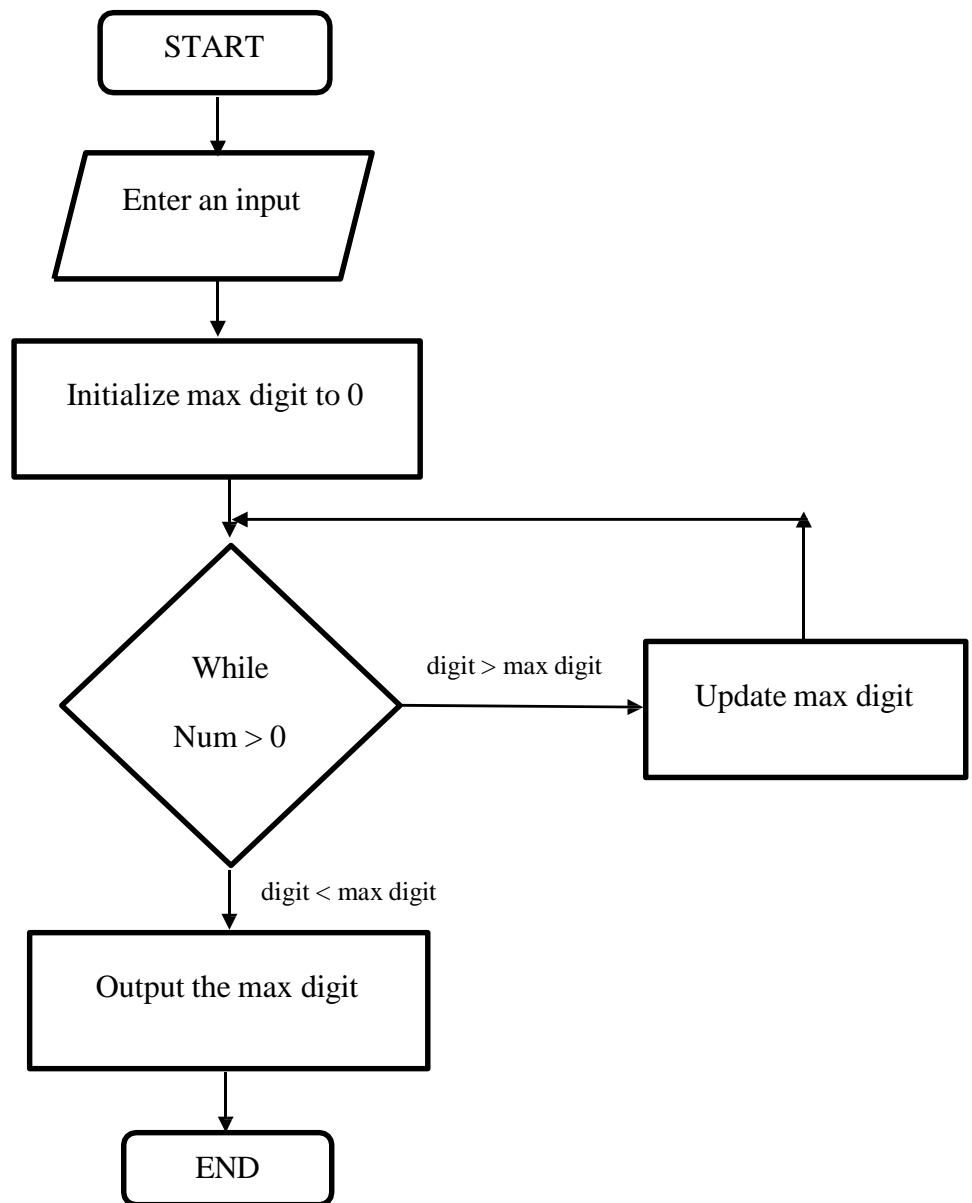
Code :

```python
def largest_digit(n):
    max_digit = 0
    while n > 0:
        digit = n % 10
        if digit > max_digit:
            max_digit = digit
        n = n // 10
    return max_digit

num = int(input("Enter a number: "))
print("The largest digit is:", largest_digit(n))
```

Output :

Result :

Date :

Exp.no: 07

Develop a Python program to display only the positive elements of the list.

Aim :

To develop a Python program to display only the positive elements of the list.

Algorithm :

STEP 1: Start

STEP 2: Get a list of numbers from the user and store it in a variable called num_list

STEP 3: Convert the input string into a list of integers using the map function and split method
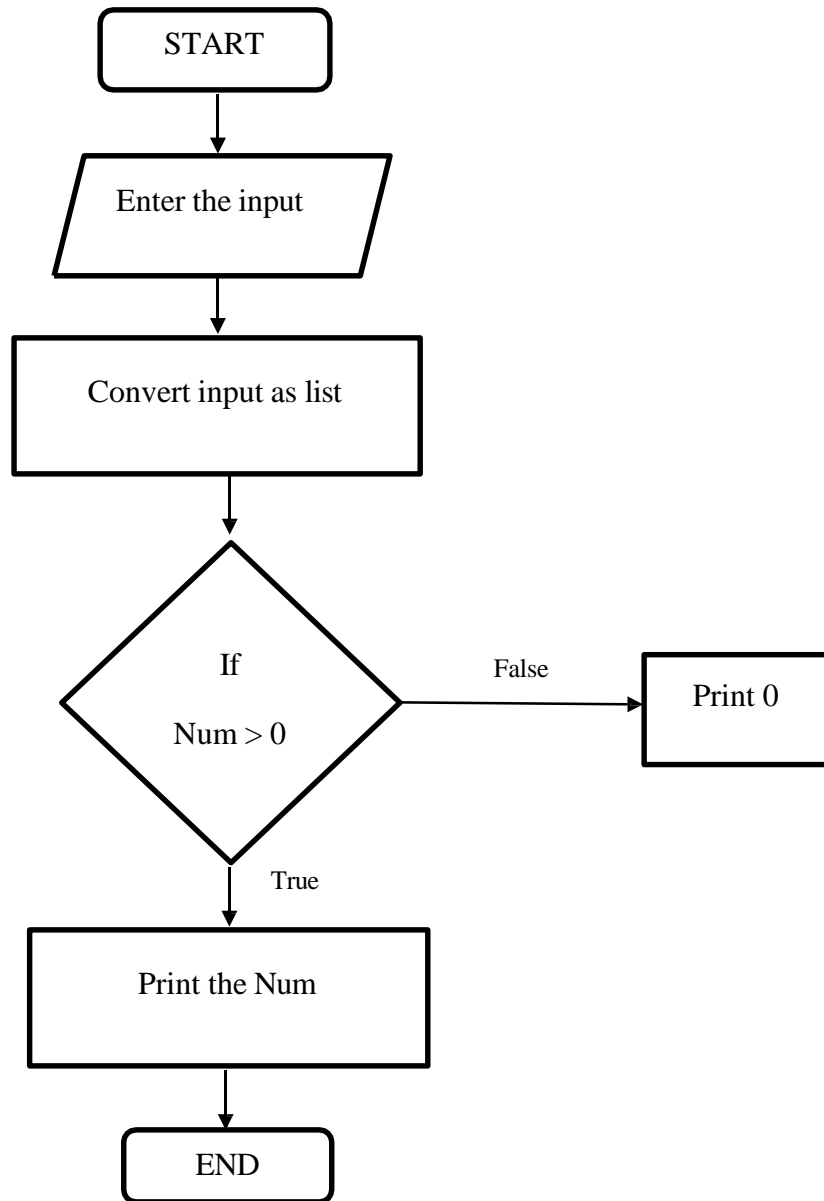
STEP 4: Create an empty list called positive_nums

STEP 5: For each number in num_list, do the following: a. If the number is greater than 0, append it to the positive_nums list

STEP 6: Print the positive_nums list

STEP 7: End

Flow Chart :

```
                        ┌─────────────┐
                        │    START    │
                        └──────┬──────┘
                               │
                               ▼
                       ╱─────────────────╲
                      ╱   Enter the input  ╲
                      ╲                    ╱
                       ╲──────────────────╱
                               │
                               ▼
                      ┌──────────────────┐
                      │ Convert input as │
                      │       list       │
                      └────────┬─────────┘
                               │
                               ▼
                             ╱   ╲
                           ╱       ╲
                         ╱    If     ╲      False
                         ╲           ╱ ──────────────►  ┌──────────┐
                         ╲  Num > 0 ╱                   │  Print 0 │
                           ╲       ╱                    └──────────┘
                             ╲   ╱
                               │
                               │ True
                               ▼
                      ┌──────────────────┐
                      │  Print the Num   │
                      └────────┬─────────┘
                               │
                               ▼
                        ┌─────────────┐
                        │     END     │
                        └─────────────┘
```

Code :

```
num_list = input("Enter a list of numbers: ")

num_list = list(map(int, num_list.split()))

print("Positive elements in the list:")

for num in num_list:

    if num > 0:

        print(num)
```

Output :

Result :

Date :

Exp.no: 08

Develop a Python program to accept any number and print it in words.

Aim :

To develop a Python program to accept any number and print it in words.

Algorithm :

STEP 1: Start

STEP 2: Define a function called printValue that takes a single argument digit

STEP 3: Inside the printValue function, use a series of if statements to check the value of digit and print the corresponding word for that digit

STEP 4: Define another function called printWord that takes a single argument N

STEP 5: Initialize a variable i to 0 and calculate the length of the input string N using the len function

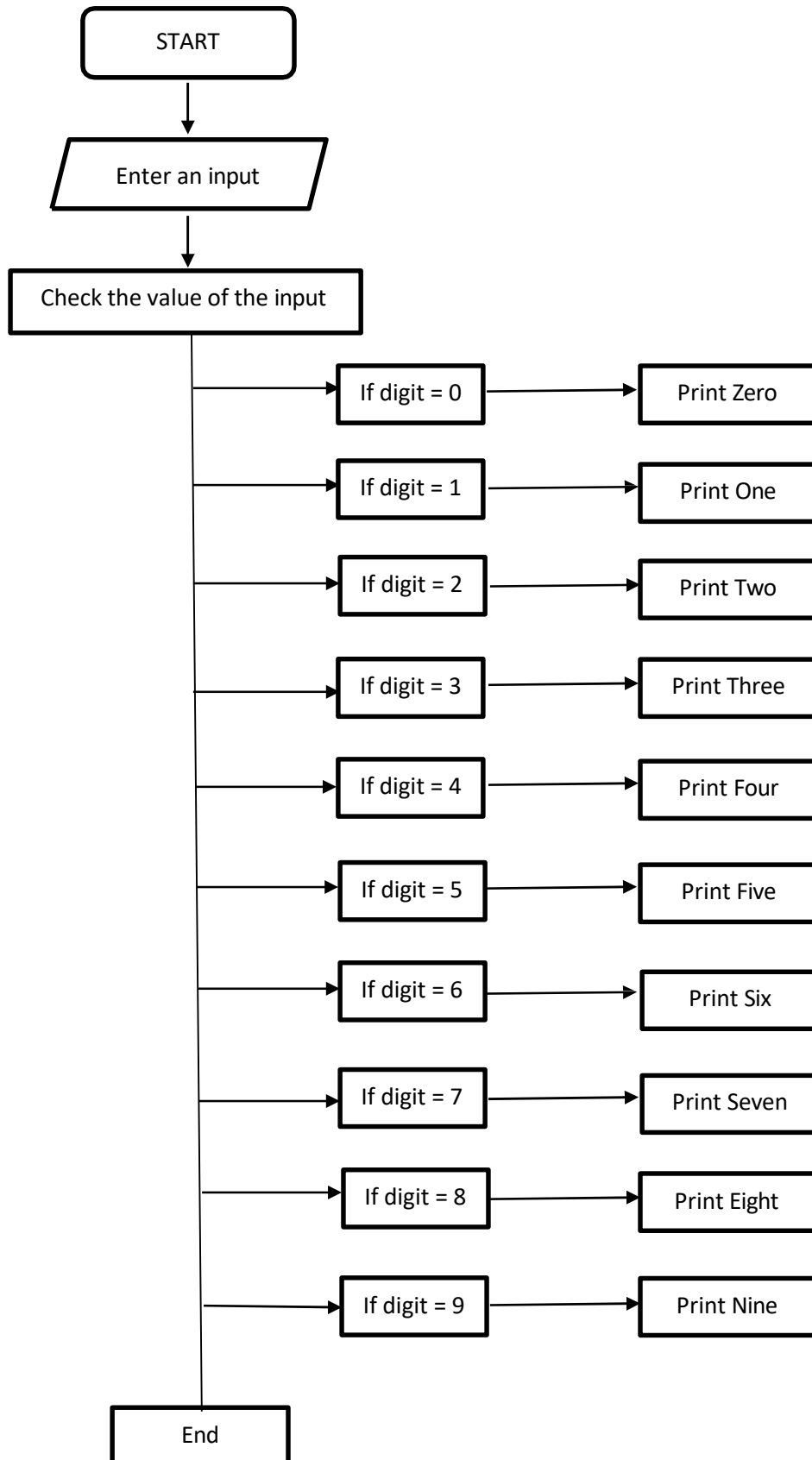**STEP 6:** Create a loop that iterates through each character in the input string N

STEP 7: Call the printValue function with the current character as an argument to print the corresponding word for that digit

STEP 8: Increment the i variable by 1 to move to the next character in the input string

STEP 9: Print the output of the printWord function

STEP 10: End

Flow Chart :

Code :

```python
def printValue(digit):

    if digit == '0':
        print("Zero ", end = " ")

    elif digit == '1':
        print("One ", end = " ")

    elif digit == '2':
        print("Two ", end = " ")

    elif digit=='3':
        print("Three",end=" ")

    elif digit == '4':
        print("Four ", end = " ")

    elif digit == '5':
        print("Five ", end = " ")

    elif digit == '6':
        print("Six ", end = " ")

    elif digit == '7':
        print("Seven", end = " ")

    elif digit == '8':
        print("Eight", end = " ")

    elif digit == '9':
        print("Nine ", end = " ")

def printWord(N):
```

31

```
    i = 0
    length = len(N)

    while i < length:

        printValue(N[i])
        i += 1

N = input("Enter a list of numbers: ")
printWord(N)
```

```
    length = len(N)
```

Output :

Result :

Date :

Exp.no: 09

Develop a Python program to subtract two matrices

Aim :

To develop a Python program to subtract two matrices.

Algorithm :

STEP 1: Initialize an empty list called matOne to store the first matrix.

STEP 2: Display "Enter 4 Elements for First Matrix: ".

STEP 3: For i in range 2, do the following:

a. Append an empty list to matOne.

b. For j in range 2, do the following:

i. Accept an integer input from the user and store it in a variable num.

ii. Append the value of num to the i-th list of matOne.

STEP 4: Initialize an empty list called matTwo to store the second matrix.

STEP 5: Display "Enter 4 Elements for Second Matrix: ".

STEP 6: For i in range 2, do the following:

a. Append an empty list to matTwo.

b. For j in range 2, do the following:

i. Accept an integer input from the user and store it in a variable num.

ii. Append the value of num to the i-th list of matTwo.

STEP 7: Initialize an empty list called matThree to store the result of the subtraction operation.

34

STEP 8: For i in range 2, do the following:

    a. Append an empty list to matThree.

    b. For j in range 2, do the following:

    i. Subtract the j-th element of the i-th list of matTwo from the j-th element of the i-th list of matOne, and store the result in a variable called sub.

    ii. Append the value of sub to the i-th list of matThree.

STEP 9: Display "Subtraction Result of Two Given Matrices is:".

STEP 10: For j in range 2, do the following:

    i. Print the j-th element of the i-th list of matThree, followed by a space and an endline character. b. Print an empty string to move to the next line.

Flow Chart :

```mermaid
flowchart TD
    START --> matOne[Initialize matOne]
    matOne --> input1[Take input for 1st Matrix]
    input1 --> matTwo[Initialize matTwo]
    matTwo --> input2[Take input for 2nd Matrix]
    input2 --> matThree[Initialize matThree]
    matThree --> subtract[Perform Subtraction of Matrices]
    subtract --> display[Display result]
    display --> END
```

START

Initialize matOne

Take input for 1st Matrix

Initialize matTwo

Take input for 2nd Matrix

Initialize matThree

Perform Subtraction of Matrices

Display result

END

Code :

```python
matOne = []
print("Enter 4 Elements for First Matrix: ")
for i in range(2):
  matOne.append([])
  for j in range(2):
    num = int(input())
    matOne[i].append(num)

matTwo = []
print("Enter 4 Elements for Second Matrix: ")
for i in range(2):
  matTwo.append([])
  for j in range(2):
    num = int(input())
    matTwo[i].append(num)

matThree = []
for i in range(2):
  matThree.append([])
  for j in range(2):
    sub = matOne[i][j] - matTwo[i][j]
    matThree[i].append(sub)

print("\nSubtraction Result of Two Given Matrices is:")
for i in range(2):
  for j in range(2):
    print(matThree[i][j], end=" ")
  print()
```

Output :

Result :

Date :

Exp.no: 10

Develop a Python program to perform matrix multiplication.

Aim :

To Develop a Python program to perform matrix multiplication.

Algorithm :

STEP 1: Define a variable MAX and initialize it to the maximum size of the matrices.

STEP 2: Define a function named printMatrix that takes three parameters: M, rowSize, and colSize. The function prints the matrix M row by row using nested loops.

STEP 3: Define a function named multiplyMatrix that takes six parameters: row1, col1, A, row2, col2, and B.

STEP 4: Create an empty matrix C of size MAXxMAX.

STEP 5: Check if the number of columns of the first matrix (col1) is equal to the number of rows of the second matrix (row2). If not, print "Not Possible" and return from the function.

STEP 6: Use nested loops to iterate over the rows and columns of the result matrix C.

STEP 7: Inside the nested loops, set the value of the element at the current position (i, j) of C to 0.

STEP 8: Use another loop to iterate over the common dimension (row2) of the two input matrices. Inside this loop, add the product of the corresponding elements of A and B to the current element of C.

STEP 9: Print the resultant matrix C using the printMatrix function.

STEP 10: In the main program, create two empty matrices A and B of size MAXxMAX.

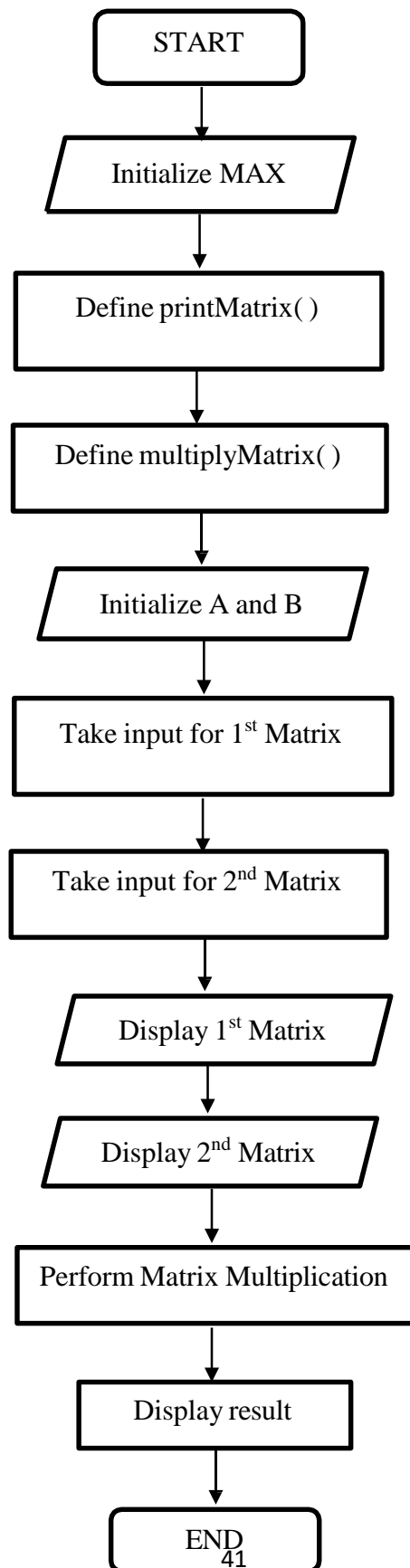**STEP 11:** Read the number of rows and columns of the first matrix A from the user and fill in the elements of A.

STEP 12: Read the number of rows and columns of the second matrix B from the user and fill in the elements of B.

STEP 13: Print the input matrices A and B using the printMatrix function.

STEP 14: Call the multiplyMatrix function with the input parameters row1, col1, A, row2, col2, and B.

Flow Chart :

```
                    ┌─────────────┐
                    │    START    │
                    └─────────────┘
                           │
                           ▼
                   ╱─────────────────╱
                  ╱  Initialize MAX ╱
                 ╱─────────────────╱
                           │
                           ▼
                  ┌──────────────────┐
                  │ Define printMatrix( ) │
                  └──────────────────┘
                           │
                           ▼
                  ┌────────────────────┐
                  │ Define multiplyMatrix( ) │
                  └────────────────────┘
                           │
                           ▼
                  ╱──────────────────╱
                 ╱  Initialize A and B ╱
                ╱──────────────────╱
                           │
                           ▼
                  ┌─────────────────────┐
                  │ Take input for 1st Matrix │
                  └─────────────────────┘
                           │
                           ▼
                  ┌──────────────────────┐
                  │ Take input for 2nd Matrix │
                  └──────────────────────┘
                           │
                           ▼
                  ╱───────────────────╱
                 ╱  Display 1st Matrix ╱
                ╱───────────────────╱
                           │
                           ▼
                  ╱────────────────────╱
                 ╱  Display 2nd Matrix ╱
                ╱────────────────────╱
                           │
                           ▼
                  ┌──────────────────────────┐
                  │ Perform Matrix Multiplication │
                  └──────────────────────────┘
                           │
                           ▼
                  ┌─────────────────┐
                  │  Display result │
                  └─────────────────┘
                           │
                           ▼
                    ┌─────────────┐
                    │     END     │
                    └─────────────┘
```

41

```python
MAX = 100
def printMatrix(M, rowSize, colSize):
    for i in range(rowSize):
        for j in range(colSize):
            print(M[i][j], end = " ")
        print()
def multiplyMatrix(row1, col1, A, row2, col2, B) :
    C = [[0 for i in range(MAX)]
         for j in range(MAX)]
    if (row2 != col1) :
        print("Not Possible")
        return

    for i in range(row1):
        for j in range(col2):
            C[i][j] = 0
            for k in range(row2):
                C[i][j] += A[i][k] * B[k][j]

    print("Resultant Matrix: ")
    printMatrix(C, row1, col2)

if __name__ == "__main__" :

    A = [[0 for i in range(MAX)]
         for j in range(MAX)]
    B = [[0 for i in range(MAX)]
         for j in range(MAX)]

row1 = int(input("Enter the number of rows of First Matrix: "))
col1 = int(input("Enter the number of columns of First Matrix: "))

print("Enter the elements of First Matrix: ")
for i in range(row1) :
```

```python
        for j in range(col1) :
            A[i][j] = int(input("A[" + str(i) +"][" + str(j) + "]: "))

row2 = int(input("Enter the number of rows of Second Matrix: "))
col2 = int(input("Enter the number of columns of Second Matrix: "))

print("Enter the elements of Second Matrix: ")
for i in range(row2) :
        for j in range(col2) :
            B[i][j] = int(input("B[" + str(i) +"][" + str(j) + "]: "))

print("First Matrix: ")
printMatrix(A, row1, col1)

print("Second Matrix: ")
printMatrix(B, row2, col2)

multiplyMatrix(row1, col1, A, row2, col2, B)
```

Output :

Result :