|  |  |
| --- | --- |
| EX.NO: 1 | **TO COMPUTE THE GCD OF TWO NUMBERS** |
| DATE: |

**AIM**

To Compute the GCD of two numbers

**ALGORITHM:**

STEP 1: Start the Program

STEP 2: Get the two input variables and assume as a and b.

STEP 3: Implement while condition, and the condition is true,

And execute the following steps under the condition.

STEP 4: In given LCM method, find the Common element and

Find the largest common element.

STEP 5: Largest element is greatest common divisor.

STEP 6: Execute the program.

**TO COMPUTE THE GCD OF TWO NUMBERS**

A= int(input(“enter 1st number:”))

B=int(input(“enter 2nd number:”))

A=C

B=D

While (num2!=0)

Temp=B

B=A%B

A=temp

Gcd=A

Print(“GCD IS”, Gcd)

**OUTPUT**

Enter 1st number : 26

Enter 2nd number : 12

GCD is 2

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 2 | **FIND SQUARE ROOT OF NUMBER** |
| DATE: |

**AIM**

To find the Square root of Number

**ALGORITHM**

STEP 1: Start the program

STEP 2: Import math function for finding square root of number.

STEP 3: Get new input for finding square root.

STEP 4: Assume the math function for new input.

STEP 5: Print the square root value.

STEP 6: Execute the program

**FIND SQUARE ROOT OF NUMBER**

Import math

Number=int(input(enter a number: ))

Sqrt=math.sqrt(number)

Print=(“square root:”, Sqrt)

**OUTPUT**

Enter a number : 81

Square root: 9

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 3 | **FIND EXPONENTATION OF GIVEN POSITIVE NUMBER** |
| DATE: |

**AIM**

To find Expoonentation of given positive number.

**ALGORITHM**

Step1: Start the program

Step2: Import the Math

Step3: Enter the input value

Step 4: Print and get exponential value

Step 5: Stop the program.

**FIND EXPONENTATION OF GIVEN POSITIVE NUMBER**

Number=int (input(“please enter any positive integer:”))

Exponent=int(input (“please enter exponent value:”))

Power=1

For I in range (1, exponent+1):

Power=power \* number

Print (“The result of {0} power {1}={2}”. Format (number,

exponent,power))

**OUTPUT**

Please enter any positive integer: 2

Please enter Exponent value:2

The Result of 2 Power 2= 4

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 4 | **TO PERFORM LINEAR SEARCH FROM THE LIST OF ELEMENT** |
| DATE: |

**AIM**

To Compute Linear search in lists using in python.

**ALOGRITHM**

Step1: Start the program

Step2: Start from the leftmost element of the list and one by one compare X with each element of the list.

Step 3: If X matches with an element return True.

Step 4: If X doesn't match with any of the elements return False.

Step 5: Stop the program.

**TO PERFORM LINEAR SEARCH FROM THE LIST OF ELEMENT**

def linear search [arr,x]:

for I in range (len(arr)):

if arr[i]=x:

return

return -1

arr=[‘t’ , ‘j’, ‘t’, ‘o’, ‘r’, ‘i’ , ‘a’, ‘l’]

x=’a’

print (“element found at index “ +str (linear search (arr,x))

**OUTPUT**

Element found at index 6

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 5 | **LIST THE FIRST N PRIME NUMBER** |
| DATE: |

**AIM**

To compute list the first N prime number.

**ALGORITHM**

STEP 1: Start the program

STEP 2: Initialize a loop starting from 2 ending at the integer value of the floor of the square root of the number.

STEP3: Check if the number is divisible by 2.

STEP 4: Repeat till the square root of the number is checked for.

STEP 5: In case the number is divisible by any of the number, the number is not prime number

STEP 6: Else it is a prime number it can be display.

STEP 7: Stop the program.

**LIST THE FIRST N PRIME NUMBER**

V= int(input (“enter range : “))

Print (“prime numbers : “ , end =” “)

For n in range (1, v):

For i in range (2,n):

if (n% i==0):

break

else:

print(n, end= “ “ )

**OUTPUT**

Enter range :2

Prime numbers : 1

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 6 | **FIND THE MAXIMUM OF A LIST OF NUMBERS** |
| DATE: |

**AIM**

To Find the maximum of a list of number

**ALGORITHM**

Step1: Start the program

Step2: Getting the input values in numbers as list

Step 3: Finding the maximum value of given list

Step 4: Print the result of maximum value of numbers

Step 5: Stop the program.

**FIND THE MAXIMUM OF A LIST OF NUMBERS**

List 1=[ ]

Num= int (input(“ Enter number of elements in list: “))

For I in range (1, num +1):

Ele = int (input(“ Enter elements :” ))

List 1.append (ele)

Print(“ The Largest is : “, max(list 1))

**OUTPUT**

Enter number of elements in list : 12

Enter elements : 24

Largest element is : 24

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 7 | **IMPLEMENTATION INSERTION SORT** |
| DATE: |

AIM

To Implementation insertion sort.

ALGORITHM

Step 1: Start the program

Step 2: Getting the values in array

Step 3: Making the looping for given array values by using

For and while loop

Step 4: Looping the function and sorting the given list

Step 5: print the list of sorting values

Step 6: Stop the program

**IMPLEMENTATION INSERTION SORT**

def insertionSort(arr):

for i in range(1, len(arr)):

key=arr[i]

j=i-1

while j>=0 and key < arr[j]:

arr[j+1]=arr[j]

j-=1

arr[j+1]=key

arr=[12,11,13,5,6]

insertionSort(arr)

for i in range(len(arr)):

print("% d" % arr[i])

OUTPUT

5

6

11

12

13

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 8 | **REMOVE ALL THE DUPLICATE ELEMENTS IN LIST** |
| DATE: |

AIM

To remove all the duplicate elements in list

ALGORITHM

Step 1: Start the program

Step 2: Create a list of numbers

Step 3: Iterate each element in list and add them in variable total

Step 4: Print the new list after removing all even number

Step 5: Stop the program.

**REMOVE ALL THE DUPLICATE ELEMENTS IN LIST**

Test\_list=[1,3,5,6,3,5,6,1]

Print(“ The original list is:” +str(Test\_list))

res=[]

[res. append (x) for x in test\_list if x not in res]

Print (“The List after removing duplicates : “ +strr(res))

**OUTPUT**

Original list is : [1,3,5,6,3,5,6,1]

List after removing duplicates: [1,3,5,6]

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO:9 | **IMPLEMENT A PYTHON PROGRAM FIND THE MOST FREQUENT WORDS IN A TEXT READ FROM A FILE** |
| DATE: |

**AIM**

To compute implement a python program find the most frequent words in a text read from a file.

**ALGORITHM**

\ Step 1: Start the program

Step 2: Take the content the file as a input

Step 3: Save each word in llist after removing spaces and punctutation from the input string

Step 4: Finding the Frequency of each word

Step 5: Print the word which has maximum frequently

Step 6: Stop the program.

**IMPLEMENT A PYTHON PROGRAM FIND THE MOST FREQUENT WORDS IN A TEXT READ FROM A FILE**

File = open (“ gfg . txt” , “r”)

Frequent\_word =” “

Frequency =0

Words = [ ]

For line in file:

Line\_word = line . lower( ) . replace(‘,’ , ‘ ‘) . replace(‘ , ‘ , ‘ ‘ ). Split(“ “);

For w in line\_word:

Words.append(w)

For I in range (0, len (words)):

Count = 1;

For j in range (i+1, len (words)

If (words [i] ==words [j]):

Count = count + 1;

If (count > frequency ):

Frequency =count :

Frequency\_word = words[i];

Print(“Most repeated word : “ + frequent\_word)

Print(“Frequency : “ + str (frequency))

File.close ( ):

**OUTPUT**

Most repeated word : well

Frequency :3

**RESULT**

The above program is execute successfully

|  |  |
| --- | --- |
| EX.NO: 10 | **SIMULATE BOUNCING BALL USING PYGAME** |
| DATE: |

**AIM**

To simulate bouncing ball using pygame

**ALGORITHM**

Step 1: Start the program

Step 2: Setting the size, speed, background

Step 3: Designing the bouncing ball in display for playing

Game in pygame

Step 4: looping the gaming method in game and setting

Looping for width and height

Step 5: Stop the grogram

**SIMULATE BOUNCING BALL USING PYGAME**

import sys, pygame

pygame.init()

size = width, height = 800, 400

speed = [1, 1]

background = 255, 255, 255

screen = pygame.display.set\_mode(size)

pygame.display.set\_caption("Bouncing ball")

ball = pygame.image.load("ball.png")

ballrect = ball.get\_rect()

while 1:

for event in pygame.event.get():

if event.type == pygame.QUIT:

sys.exit()

ballrect = ballrect.move(speed)

if ballrect.left < 0 or ballrect.right > width:

speed[0] = -speed[0]

if ballrect.top < 0 or ballrect.bottom > height:

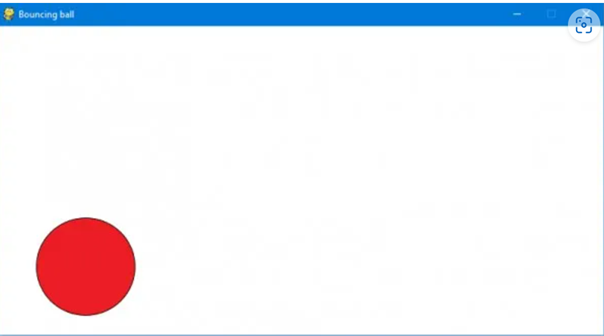
speed[1] = -speed[1]

screen.fill(background)

screen.blit(ball, ballrect)

pygame.display.flip()

**OUTPUT**



**RESULT**

The above program is execute successfully