Ex. No:	1	FIBONACCI SERIES
Date:		

ALGORITHM:

Step1:Prompt the user to enter the number of terms and store it in the variable no_turns.

Step2;store it in the variable no_turns.

Step3:Initialize variables n1 and n2 to 0 and 1, respectively.

Step4:Initialize count to 0.

Step5:Check if no_turns is less than or equal to 0. If true, print "Please enter a positive number" and exit.

Step6;Check if no_turns is equal to 1. If true, print n1 and exit.

Step7:Enter a loop while no_turns is greater than count.

Step8:Print the value of n1.

Step9:Calculate the next term by adding n1 and n2 and store it in the variable next.

Step10:Update n1 to n2 and n2 to next.

Step11:ncrement count by 1.

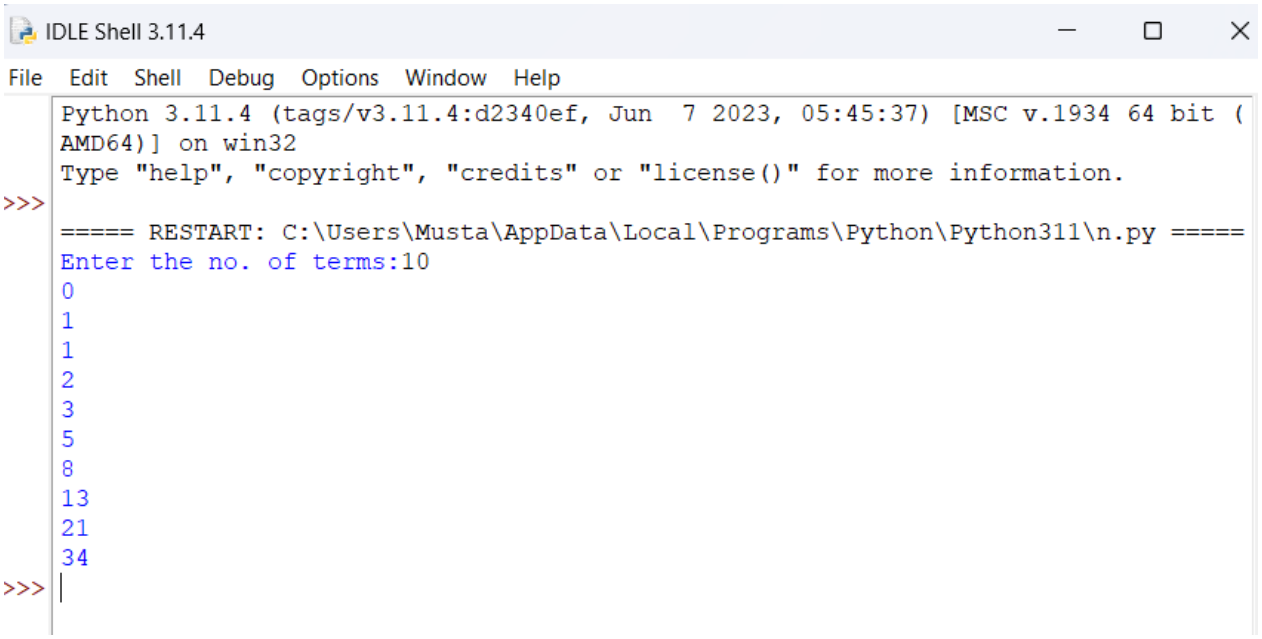
Step12:End the loop.

Step13:Stop

PROGRAM CODE:

```
no_turns  = int(input("Enter the no. of terms:"))
n1 = 0
n2 = 1
count = 0
if no_turns<= 0:
  print("Please enter the positive number")
elif no_turns == 1:
  print(n1)
else:
  while no_turns > count:
    print(n1)
    next = n1+n2
    n1 = n2
    n2 = next
    count+=1
```

OUTPUT:


IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Musta\AppData\Local\Programs\Python\Python311\n.py =====
Enter the no. of terms:10
0
1
1
2
3
5
8
13
21
34
>>>

Ex. No:	2	MAXIMUM OF THREE NUMBERS
Date:		

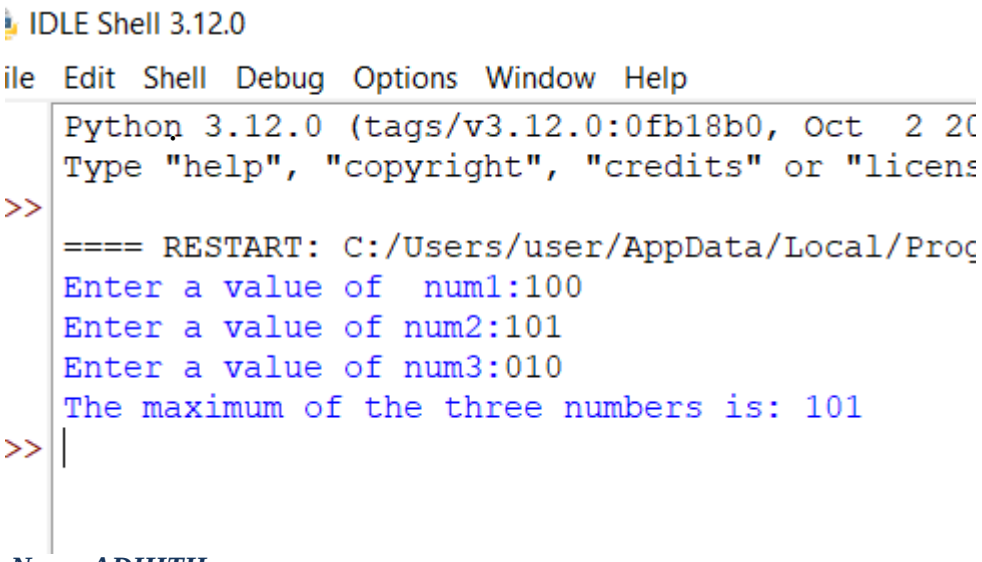
ALGORITHM:

Step1:Prompt the user to enter three numbers and store them in variables num_1, num_2, and num_3.
Step2:Check if num_1 is greater than num_2 or num_3. If true, print num_1 followed by "is the greatest".
Step3:If the previous condition is false, check if num_2 is greater than num_1 or num_3. If true, print num_2 followed by "is the greatest".
Step4:If both previous conditions are false, print num_3 followed by "is the greatest".

PROGRAM CODE:

```
num_1 = int(input("Enter a number"))
num_2 = int(input("Enter a number"))
num_3 = int(input("Enter a number"))
if num_1>num_2 or num_1>num_3:
print(num_1,"is the greatest")
elif num_2>num_1 or num_2>num_3:
  print(num_2,"is the greatest")
else:
  print(num_3,"is the greatest")
```

OUTPUT:



Ex. No:	3	LEAP YEAR OR NOT
Date:		

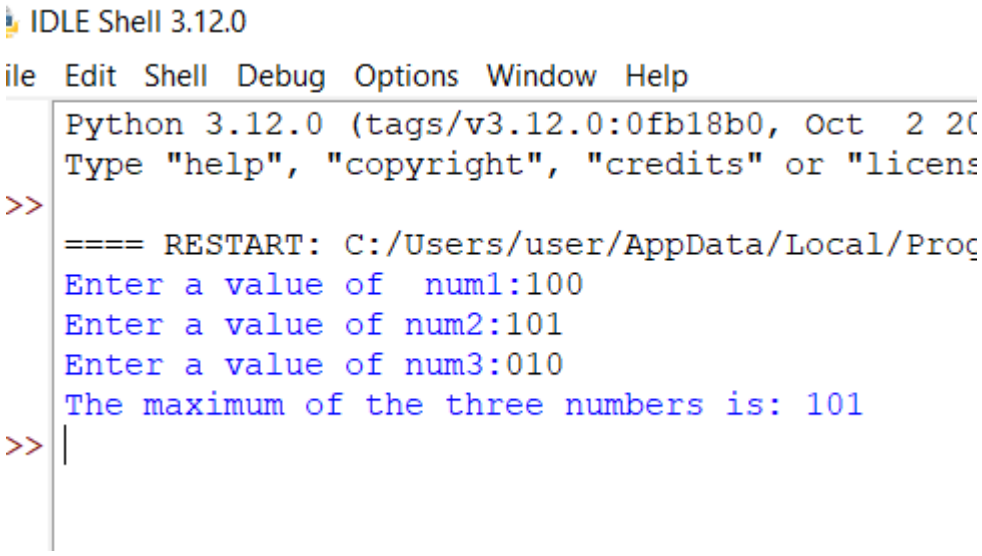
ALGORITHM:

Step1:Prompt the user to enter a year and store it in the variable year.
Step2:Check if year is divisible by 4 using the condition year % 4 == 0.
Step3:if true then go to next next step
Step4:Check if is divisible by 400 using the condition year % 100 == 0.
Step5:If true go to nexyt step.
Step6:Check if is divisible by 100 using the condition year % 400 == 0.
Step7:If true, print "It is a leap year" and exit.
Step8:If false again repeat the process
Step9:Print "It is not a leap year" and exit.
Step10:Print "It is not a leap year" and exit.

PROGRAM:

```
year = int(input("Enter a year:"))
if year%4 == 0:
  print("It is a leap  year")
elif year%400 == 0:
print("It is a leap year")
elif year%100 == 0:
  print("It is not a leap year")
else:
  print("It is not a leap year")
```

OUTPUT:



Ex. No:	4	SUM OF NUMBERS IN A LIST
Date:		

ALGORITHM:

Step1:Start

Step2create the list and insert the values .
Step3:compute the list by using the for loop and append into the the list
Step4:And now by using the sum method calculate the sum of the list .
Step5:print the list
Step6:Stop
.

PROGRAM:

```
lst = []
number_of_elements = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
print("The sum of the given list",lst,"is",sum(lst))
```

OUTPUT:

```
>>>
================================================
The sum of all numbers in the list is: 100
>>>
```

Ex. No:	5	FIND ELEMENT IN A LIST USING BINARY SEARCH
Date:		

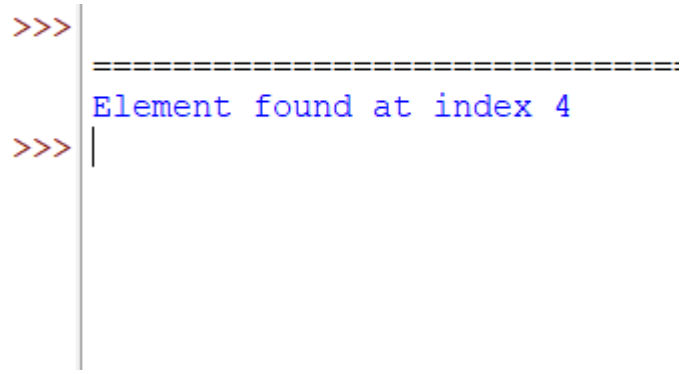
ALGORITHM:

Step1:Start
Step2:define the function
Step3:set the low value as 0
Step4:set the high value as length of the list -1
Step5:set the mid value as low+high and divided by 2
Step6:if mid value is equal to the token value hence we got the element
Step7:otherwise if low add 1 to mid and if high sub 1 to mid
Step8:Stop

PROGRAM:

```
def binary_search(alist, token):
    low = 0
      high = len(alist) - 1
    mid = (low+high) // 2
    if alist[mid] == token:
    print(token)
      return
    elif alist[mid] < token:
      low = mid + 1
    else:
      high = mid - 1

token = int(input("Enter a number:"))
alist = [1,3,5,9]
binary_search(alist,token)
```

OUTPUT:



Name:AJAY.S ***Branch:*** *B. Tech AI&DS*

Ex. No:	6	MULTIPLY ALL NUMBERS IN A LIST
Date:		

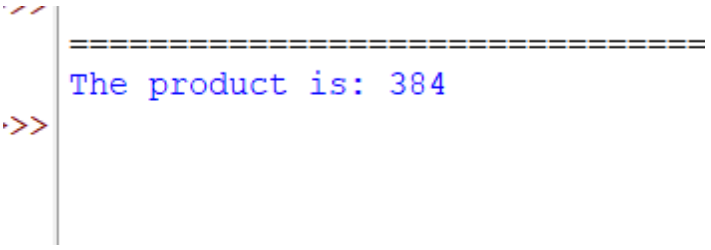
ALGORITHM:

Step1:Start
Step2:create the list and insert the element using append functionStep3:take the other variable as 1
Step4:Compute the loop of created list and store the value in the a
Step5:then take the condition x*a
Step6:print x value
Step7:Stop

PROGRAM:

```
lst = []
number_of_elements = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
x=1
for a in lst:
  x=x*a
print("The multiplication of the list",lst,"is",x)
```

OUTPUT:



Ex. No:	7	LARGEST NUMBER IN A LIST
Date:		

ALGORITHM:

Step1:Start
Step2:create the list and insert the value in the list
Step3:and insert the value by using the append function
Step4:print the list
Step5:and to display the largest number in the list by using the max method
Step6: Stop

PROGRAM:

```
lst = []
number_of_elements = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
max_list = print("The largest number in a list",lst,"is",max(lst))
```

OUTPUT:

```
>>>
============================
The largest number is: 20
>>>
```

Ex. No:	8	PRIME OR NOT
Date:		

ALGORITHM:

Step1:Start
Step2:Get the input from the user and store in num_1
Step3:set the condition input is equal to 2
Step4:print the given number is prime
Step5:and if input divided by 2 is equal to 0
Step6:print the given number is not a prime number
Step7:otherwise the given number is a prime number
Step8:Stop

PROGRAM:

```
num_1 = int(input("Enter a number:"))
if num_1 == 2:
  print("The given number is prime number")
  pass
elif num_1%2 == 0:
  print("The given number is not a prime number")
else:
print("The given number is prime number")
```

OUTPUT:

```
==============================
17 is a prime number
>>>
```

Ex. No:	9	ODD OR EVEN
Date:		

ALGORITHM:

Step1:Start
Step2:Get the input from the user and store the value in num_1
Step3:divide the input with 2 and if it is equal to 0
Step4:print the nimber is even
Step5:otherwise the given number is odd
Step6:Stop

PROGRAM:

```
num_1 = int(input("Enter a number:"))
if num_1%2 == 0:
  print("The given number is even")
else:
  print("The given number is odd")
```

OUTPUT:

```
====================
14 is even
>>>
```

Ex. No:	10	FIND TWO STRINGS ARE ANAGRAM
Date:		

ALGORITHM:

Step1:Start
Step2:get the input from the user and store in the str_1
Step3:get the input from the user and store in the str_2
Step4:set the str_1 and str_2 with the lower() function
Step5:check the condition using nested if and check the other conditions
Step6:if sorted str_1 is equal to sorted str_2 it is anagram
Step7:otherwise not an anagram
Step8:Stop

PROGRAM:

```
str_1 = input("Enter a string:")
str_2 = input("Enter a string:")
str_1 = str_1.lower()
str_2 = str_2.lower()
if len(str_1) == len(str_2):
  if sorted(str_1) == sorted(str_2):
  print(str_1,"and",str_2,"are anagram")
  else:
    print(str_1,"and",str_2,"are not anagram")
else:
  print(str_1,"and",str_2,"are not anagram")
```

OUTPUT:

```
==================
True
False
>>>
```

Ex. No:	11	Check string is Palindrome or Not
Date:		

ALGORITHM:

Step1:Start
Step2:Get the input from the user for string
Step3:set the input string is equal to the sliced string
Step4:then print given string is a palindrome
Step5:otherwise it is a palindrome
Step6:Stop

PROGRAM:

```
str = input("Enter a string:")
str_1 = str[::-1]
if str == str_1:
  print("The given string is a palindrome")
else:
  print("The given string is not a palindrome")
```

OUTPUT:

```
>
========================================
madam is a Palindrome String
>
```

Ex. No:	12	ADDITION OF TWO MATRIX
Date:		

ALGORITHM:

Step1:Start

Step2:Initialize an empty list result to store the resultant matrix.

Step3:Iterate over each row i in mat1 and mat2.

Step4:Initialize an empty list row to store the current row of the resultant matrix.

Step5: Iterate over each column j in mat1[i] and mat2[i].

Step6:Add the corresponding elements from mat1 and mat2 and append the sum to row.

Step7:Append row to result.

Step8:Return result as the final result.

Step9:Stop

PROGRAM:

```
from re import sub
def sub_matrices(mat1, mat2): result =
    []
    for i in range(len(mat1)): row
        = []
        for j in range(len(mat1[0])):
            row.append(mat1[i][j] + mat2[i][j])
        result.append(row) return
    result
matrix1 = [[1, 2, 3], [4, 5, 6]]
matrix2 = [[7, 8, 9], [10, 11, 12]]
result_matrix = sub_matrices(matrix1, matrix2)
print("Resultant Matrix after addition:")
for row in result_matrix: print(row))
```

OUTPUT:

```
========================
Result matrix is:
[6, 8]
[10, 12]
>>>
```

Ex. No:	13	CLONING A LIST
Date:		

ALGORITHM:

Step1:Start
Step2:first import the copy module
Step3:Get the input from the user and store in lst
Step4:at the third varible lst_2 set copy of list
Step5:print lst_2
Step6:The element inside the lst will be copied to lst_2
Step7:Stop

PROGRAM:

```
import copy
lst = int(input("Enter a number"))
lst_2 = copy.copy(lst)
print("The element in the lst_2 is",lst_2)
```

OUTPUT:

```
>>>
==========================================
Original List: [1, 2, 3]
Cloned List: [1, 2, 3]

Original List after changing cloned list:
[1, 2, 3]
Cloned List: [5, 2, 3]
>>>
```

Ex. No:	14	SUBTRACTION OF TWO MATRICES
DATE:		

ALGORITHM:

Step1:Start

Step2:Initialize an empty list result to store the resultant matrix.

Step3:Iterate over each row i in mat1 and mat2.

Step4:Initialize an empty list row to store the current row of the resultant matrix.Step5:

Iterate over each column j in mat1[i] and mat2[i].

Step6:Sub the corresponding elements from mat1 and mat2 and append the sub to row.

Step7:Append row to result.

Step8:Return result as the final result.

Step9:Stop

PROGRAM:

```
from re import sub
def sub_matrices(mat1, mat2): result = []
    for i in range(len(mat1)): row = []
        for j in range(len(mat1[0])):
            row.append(mat1[i][j] - mat2[i][j])
        result.append(row) return result
matrix1 = [[1, 2, 3], [4, 5, 6]]
matrix2 = [[7, 8, 9], [10, 11, 12]]
result_matrix = sub_matrices(matrix1, matrix2)
print("Resultant Matrix after addition:")
for row in result_matrix: print(row)
```

OUTPUT:

```
>>>
======================
Matrix X:
[1, 2]
[3, 4]
Matrix Y:
[5, 6]
[7, 8]
Result matrix:
[-4, -4]
[-4, -4]
>>>
```

Ex. No:	15	FIRST N NUMBERS DIVISIBLE BY 5
DATE:		

ALGORITHM:

Step1:Start
Step2:Get the input from the user and store it in num_1
Step3:Take the third variable x is equal to 5
Step4:Compute the loop of num_1 and store the value in i
Step5:And inside the loop check the condition if I is equal to 0 pass
Step6:or else if i is divided by 5 is equal to 0 then print i
Step7:otherwise pass
Step8:Stop

PROGRAM:

```
num_1 = int(input("Enter a number:"))
x=5
for i in range(num_1):
  if i == 0:
    pass
  elif i%5 == 0:
    print(i)
  else:
    pass
```

OUTPUT:

```
>>>
========================================
First 10 numbers divisible by 5 are:
5
10
15
20
25
30
35
40
45
50
>>>
```

Ex. No:	16	SORT ELEMENTS IN A LIST USING SELECTION SORT
DATE:		

ALGORITHM:

Step1:Start

Step2: Initialize an empty list li to store the elements.

Step3:Prompt the user to enter number_elements elements.

Step4:Iterate number_elements times:

Step5:Prompt the user to enter an element and append it to the li list.

Step6:Print the original list li.

Step7:Calculate the length of the list length using len(li).

Step8:Define a function selectionsort that takes li and length as parameters.

Step9:Within the selectionsort function:

Step10:Iterate length times using i as the loop variable.

Step11:Initialize min to li[i] and pos to i.

Step12:Iterate from i+1 to length using j as the loop variable.

Step13:If li[j] is less than min, update min to li[j] and pos to j.

Step14:Swap li[i] with li[pos].

Step15:Return the sorted list li.

Step16:Call the selectionsort function with li and length as arguments and assign the result to sorted_list.

Step17:Print the sorted list.

Step18:Stop

PROGRAM:

```
li = []
number_elements =int(input("Enter the elements:"))
for i in range(1,number_elements+1):
    elements = int(input("Enter the elements in the list:"))
    li.append(elements)
    print(li)
length  = len(li)
def selectionsort(li,length):
    for i in range(length):
```

```
        min = li[i]
        pos = i
        for j in range(i+1,length):
            if li[j] < min:
                min = li[j]
                pos = j
        li[i],li[pos] = li[pos],li[i]
    return li
print("selection sorted list is",selectionsort(li,length))
```

OUTPUT:

```
>>>
========================================
Sorted list is:  [1, 2, 3, 5, 8]
>>>
```

Ex. No:	17	STORE N NUMBER OF ELEMENT IN A LIST
DATE:		

ALGORITHM:

Step1:Start
Step2:Get the input from the user for the number of elements
Step3:Compute the loop for number of elements and store in i
Step4:Get the input from the user for the elements for the list
Step5:finally append the elements in the list
Step6:print the list
Step7:Stop

PROGRAM:

```
li = []
number_elements =int(input("Enter number of
elements:”))
for i in range(1,number_elements+1):
    elements = int(input("Enter element"))
    li.append(elements)
    print(“2 elements stored in list:”,li)
```

OUTPUT:

```
>>>
========================================
Enter number of elements: 2
Enter element: 4
Enter element: 6
2 elements stored in list: ['4', '6']
```

Ex. No:	18	LINEAR SEARCH
DATE:		

PROGRAM:

```
def linear_search(arr, target):
  for i in range(len(arr)):
      if arr[i] == target:
          return i # Return the index of the target if found
return -1  # Return -1 if the target is not found in the array
arr = []
number_elements =int(input("Enter the elements:"))
for i in range(1,number_elements+1):
    elements = int(input("Enter the elements in the list:"))
    arr.append(elements)
    print(arr)
target = int(input("Enter the target number"))
index = linear_search(arr, target)
print(f"The target {target} is found at index: {index}")
```

OUTPUT:

```
>>>
============================
Element found at index: 4
>>>
```

Ex. No:	19	SUM OF SERIES FROM 1 TO N
DATE:		

ALGORITHM:

Step1:Start
Step2:Get the input from the user and store in the variable start
Step3:Get the input from the user and store the value in end
Step4:Take the third variable x as the value as 0
Step5:Compute the loop by the start and end and store the value in i
Step6:Intialize the equation x+i and store the value in x
Step7:Print x
Step8:Stop

PROGRAM:

```
start = int(input("Enter a starting number:"))
end = int(input("Enter a ending number"))
x=0
for i in range(start,end+1):
  x= x+i
print(x)
```

OUTPUT:

```
>>>
====================================
The sum from 1 to 5 is: 15
>>>
```

Ex. No:	20	SORT N NUMBERS USING MERGE SORT
DATE:		

PROGRAM:

```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]

        merge_sort(L)
        merge_sort(R)

        i = j = k = 0

        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1

        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1

        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1
```

```
    arr = []
    number_elements =int(input("Enter the elements:"))
    for i in range(1,number_elements+1):
        elements = int(input("Enter the elements in the list:"))
        arr.append(elements)
        print(arr)
    merge_sort(arr)
print("Sorted array is:", arr)
```

OUTPUT:

```
>>>
==========
Original Numbers: [5, 2, 9, 1, 5, 6]
Sorted Numbers: [1, 2, 5, 5, 6, 9]
>>>
```

Ex. No:	21	GENERATE ALL PERMUTATIONS OF A GIVEN STRING
DATE:		

ALGORITHM:

Step1:Start
Step2:First import the permutations module
Step3:Define the function of permutation
Step4:Inside the function take the pemutation input in the string
Step5:Compute the loop by using the permutations input and store the value in p then print p
Step6:Outside the function take the input for string and at last call the function
Step7:Stop

PROGRAM:

```
from itertools import permutations

def find_permutations(str):
    perm = permutations(str)
    for p in list(perm):
        print(''.join(p))
input_str = input("Enter a string")
find_permutations(input_str)
```

OUTPUT:

```
>>>
==========
['abc', 'acb', 'bac', 'bca', 'cab', 'cba']
>>>
```

Ex. No:	22	INSERT A CARD IN A LIST OF SORTED CARDS
DATE:		

PROGRAM:

```
def insert_card(sorted_cards, new_card):
  if len(sorted_cards) == 0:
     sorted_cards.append(new_card)
     return sorted_cards

  for i in range(len(sorted_cards)):
     if new_card < sorted_cards[i]:
        sorted_cards.insert(i, new_card)
        return sorted_cards

  sorted_cards.append(new_card)
return sorted_cards
sorted_cards = [2, 4, 6, 8]
new_card = int(input("Enter a new card:"))
print(insert_card(sorted_cards, new_card))
```

OUTPUT:

```
>>>
    ====================================================
    Original Sorted Cards: [2, 5, 7, 10, 12]
    Sorted Cards After Insertion: [2, 5, 7, 8, 10, 12]
>>>
```

Ex. No:	23	SUM OF CUBES OF VALUES OF N VARIABLES
DATE:		

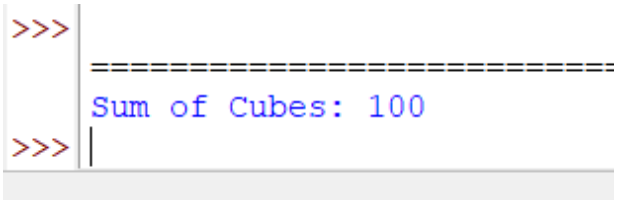
ALGORITHM:

Step1:Start
Step2:Define the function of sum of cubes values
Step3:By using the list comprehension initialize the cube x**3
Step4:And in outside the function get the input for the result
Step5:Finally print the result
Step6:Stop

PROGRAM:

```
def sum_of_cubes(*args):
    return sum(x**3 for x in args)
result = sum_of_cubes(1, 2, 3, 4)
print(result)
```

OUTPUT:



Ex. No:	24	ROOTS OF QUADRATIC EQUATION
DATE:		

ALGORITHM:

PROGRAM:

```
import cmath
def find_roots(a, b, c):
    d = (b**2) - (4*a*c)
    root1 = (-b-cmath.sqrt(d)) / (2*a)
    root2 = (-b+cmath.sqrt(d)) / (2*a)
    return root1, root2
a = int(input("Enter a number:"))
b = int(input("Enter a number:"))
c = int(input("Enter a number:"))
print(find_roots(a, b, c))
def sum_of_cubes(*args):
    return sum(x**3 for x in args)
result = sum_of_cubes(1, 2, 3, 4)
print(result)
```

OUTPUT:

```
>>>
==================================
Root 1:  (2+0j)
Root 2:  (1+0j)
>>>
```

Ex. No:	25	DIGIT AT ONE'S PLACEOF A NUMBER
DATE:		

ALGORITHM:

Step1:Start
Step2:Define the function of finding the one's place in an number
Step3:Then return the input num with the modulus of 10
Step4:At outside the function get the input from the user and store the value in number
Step5:Finally then print the number
Step6:Stop

PROGRAM:

```
def ones_place_digit(num):
  return num % 10
number = int(input("Enter a number:"))
print("The digit at one's place is:", ones_place_digit(number)))
```

OUTPUT:

```
>>>
    ================================
    Digit at one's place: 7
>>>
```

Ex. No:	26	ADDING ELEMENTS IN A LIST USING LOOPS
Date:		

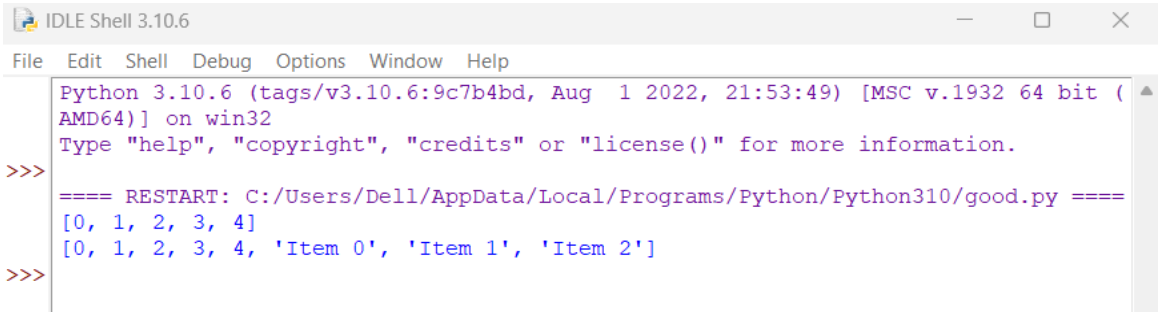
ALGORITHM:

Step1:Start
Step2:Initialize the values of the list and insert the elements in the list
Step3:Then finally print the list
Step4:Take the third variable x as the value as 0
Step5:Compute the loop of list and store the values in the i
Step6:Intialize the equation x+i and store the value in x
Step7:Print x
Step8:Stop

PROGRAM CODE:

```
lst = []
number_of_elements = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
x = 0
for i in lst:
x= x+i
print(x)
```

OUTPUT:



Ex. No:	27	Print five 1's in 5 lines
Date:		

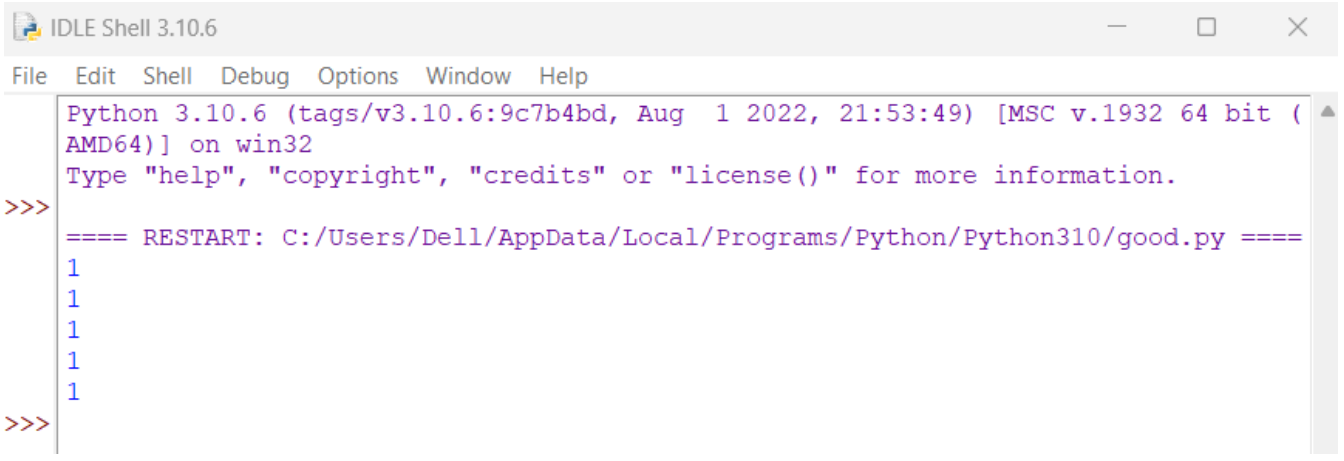
ALGORITHM:

Step1:Start
Step2:Get the input for the number of repetition
Step3:Then take the 1 in a string and with the help of repetition symbol
Step4:Stop

PROGRAM:

```
n = 5
print("1"*n)
```

OUTPUT:



Ex. No:	28	Python program to check random number in python
Date:		

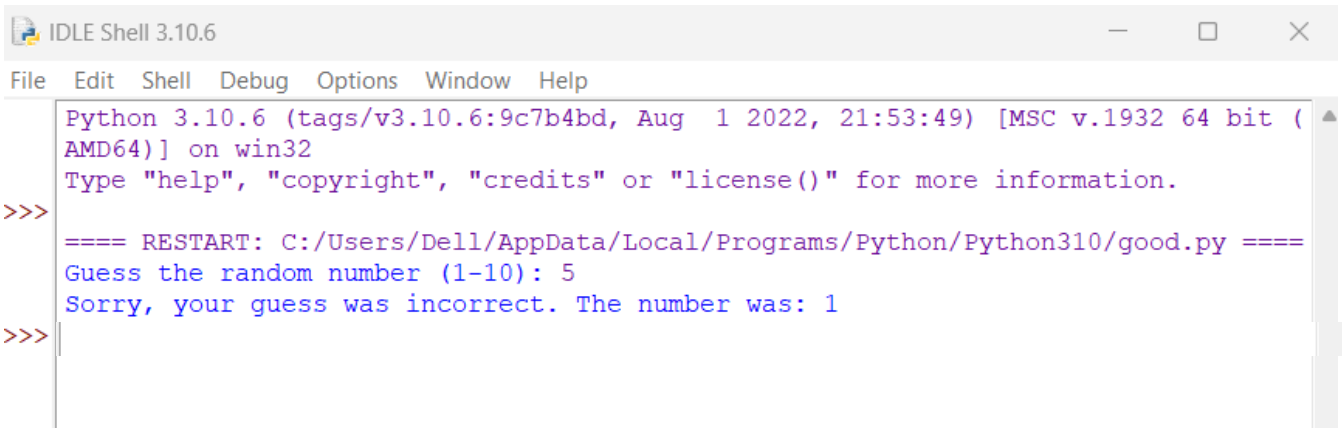
ALGORITHM:

Step1:Start
Step2:import the random module
Step3:Define the random function
Step4:At the outside the function create and initialize the element in the list
Step5:Get the input from the user for number to check
Step6:Finally call the function
Step7:Stop

PROGRAM:

```
import random
def is_random(num, lst):
    return num in lst
lst = []
number_of_elements = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
number_to_check = 5
print(is_random(number_to_check, lst))
```

OUTPUT:



\

Ex. No:	29	Python program to check squareroot of a number
Date:		

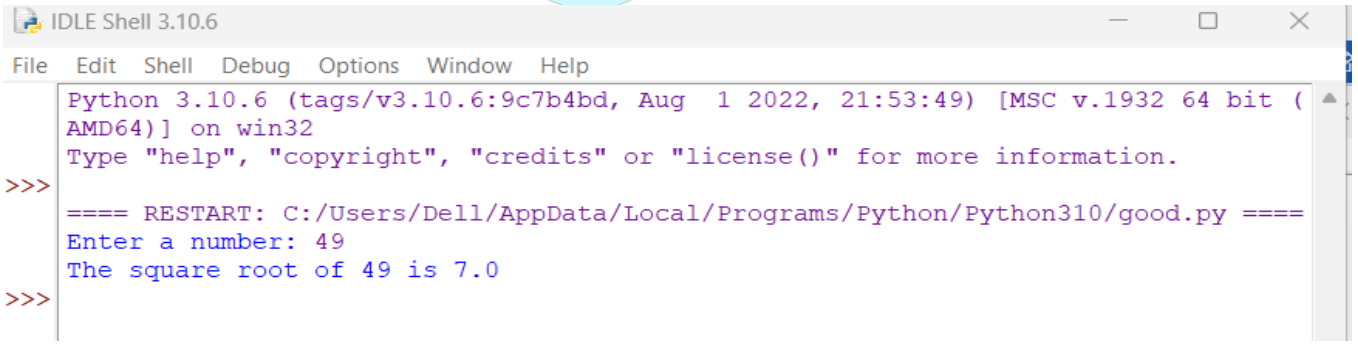
ALGORITHM:

Step1:Start
Step2:Get the input from the user to check the squareroot of the number
Step3: Initialize the input with the double modulus of 0.5
Step4:Stop

PROGRAM:

```
num = int(input("Enter a number to check the square root:"))
print("The square root of the ",num,"is",(num)**0.5)
```

OUTPUT:



Ex. No:	30	Python program to convert kilometers into miles
Date:		

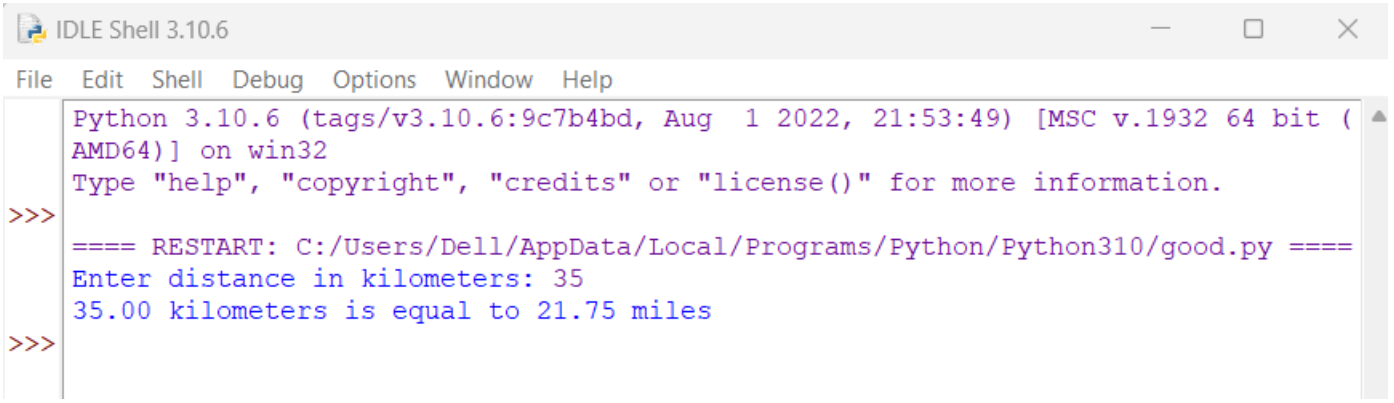
ALGORITHM:

Step1:Start
Step2:Get the input from the user for converting kilometers into miles
Step3:Then multiply the given input to the 0.62137119
Step4:Stop

PROGRAM:

```
num = int(input("Enter a kilometer:"))
print("The given kilometer",num,"to miles is",(num*0.62137119))
```

OUTPUT:



Ex. No:	31	CONVERT CELCIUS TO FAHRENHEIT
Date:		

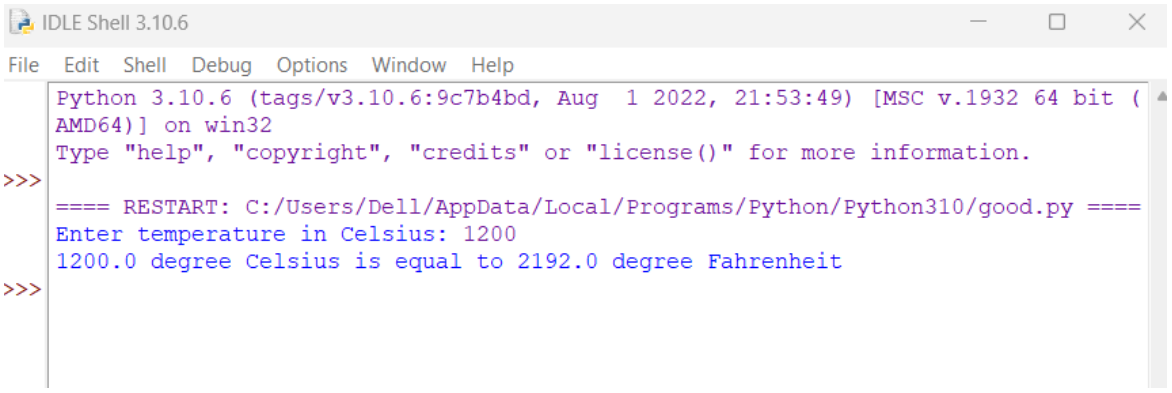
ALGORITHM:

Step1:Start
Step2:Get the input from the user to convert the celcius to fahrenheit
Step3: Initialize the input multiply with the 9/5 and add whole equation with the 32
Step4:Stop

PROGRAM:

```
num = int(input("Enter the celcius:"))
print("The given celcius",num,"into fahrenheit is",(num*9/5)+32)
```

OUTPUT:

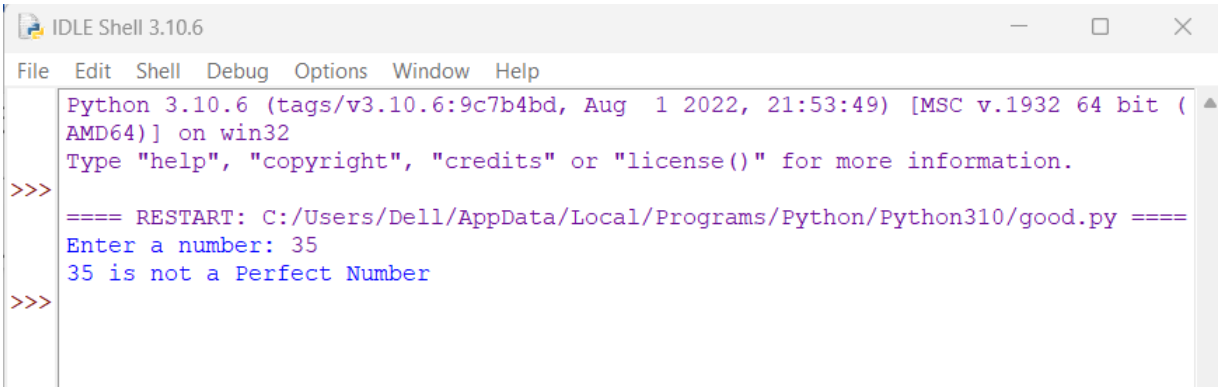


Ex. No:	32	Python program to identify whether entered num is a perfect number
Date:		

PROGRAM:

```
num=int(input("Enter the number: "))
sum_v=0
for i in range(1,num):
    if (num%i==0):
        sum_v=sum_v+i
if(sum_v==num):
    print("The entered number is a perfect number")
else:
print("The entered number is not a perfect number")
```

OUTPUT:



Ex. No:	33	covert decimal into binary in python
Date:		

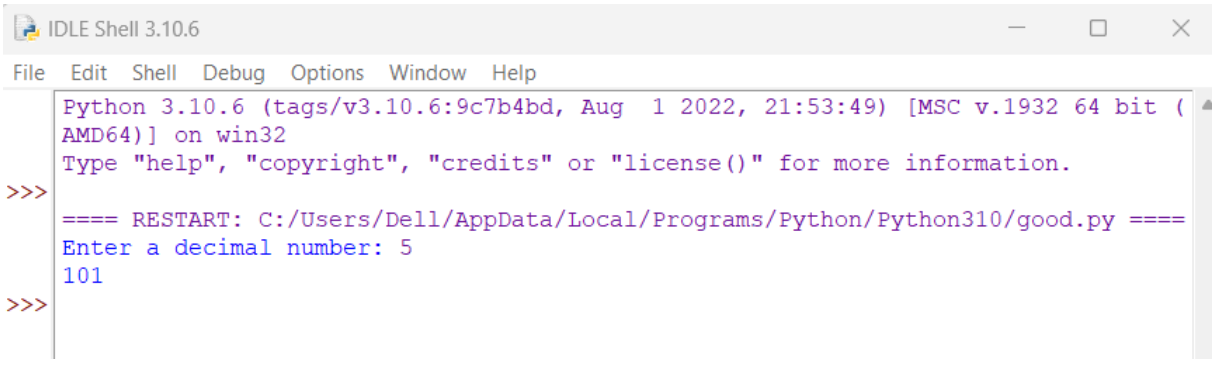
ALGORITHM:

Step1:Start
Step2:Get the input from the user to convert the decimal into the binary
Step3: And by using the slicing operator get the binary value
Step4:Stop

PROGRAM:

```
n = int(input("Enter a decimal number:"))
print(oct(n)[2:])
```


OUTPUT:

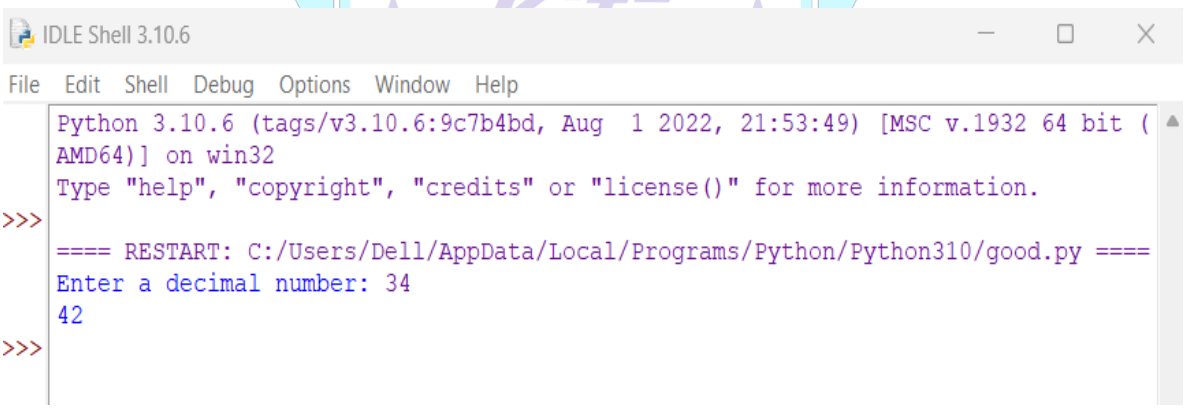


Ex. No:	34	convert decimal to octal in python
Date:		

PROGRAM:

```
n = int(input("Enter a decimal number:"))
print(oct(n)[2:])
```

OUTPUT:

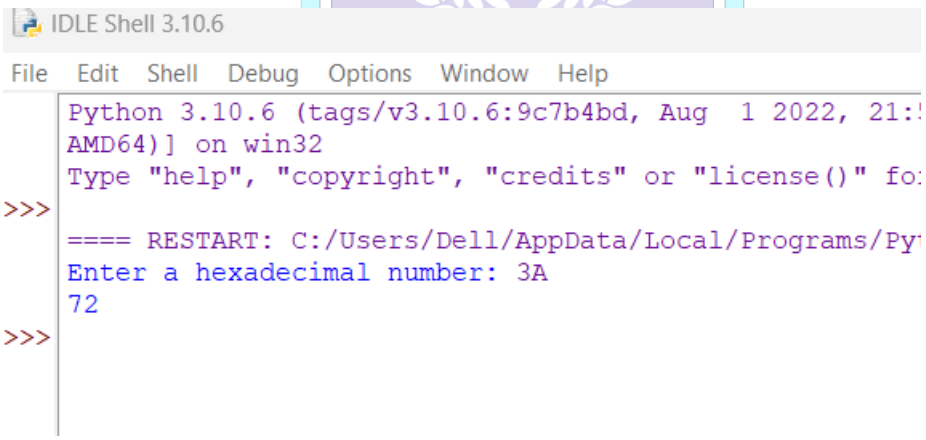


Ex. No:	35	convert from hexadecimal to octal in python
Date:		

PROGRAM:

```
print("Enter Hexadecimal Number: ")
hexa_dec_no = input()
octal_no = int(hexa_dec_no, 16)
octal_no = oct(octal_no)
print("\nEquivalent Octal Value = ", octal_no)
```

OUTPUT:

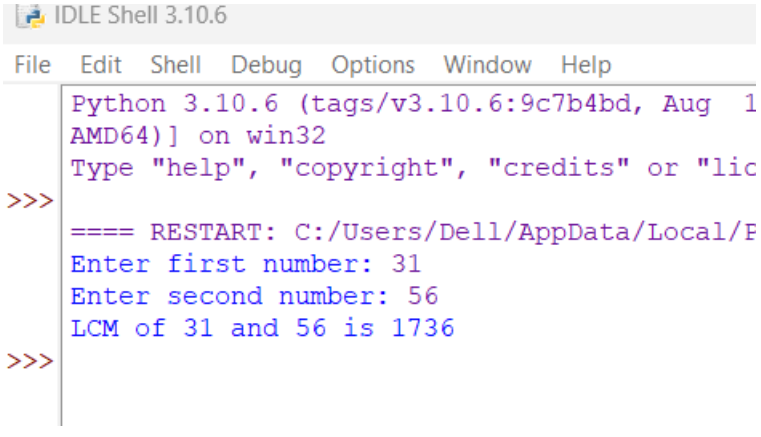


Ex. No:	36	program to find LCM of two numbers
Date:		

PROGRAM:

```
def compute_lcm(x, y):

   # choose the greater number
   if x > y:
      greater = x
   else:
      greater = y

   while(True):
      if((greater % x == 0) and (greater % y == 0)):
         lcm = greater
         break
      greater += 1

   return lcm

num1 = int(input("Enter the first number:"))
num2 = int(input("Enter the second the number:"))

print("The L.C.M. is", compute_lcm(num1, num2)))
```

OUTPUT:

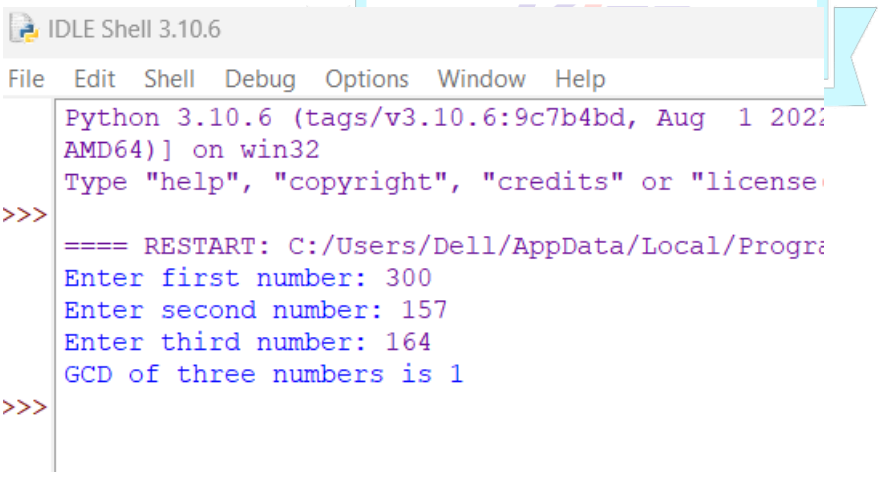


Ex. No:	37	program to find GCD of three numbers
Date:		

PROGRAM:

```
# Take inputs
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
num3 = int(input("Enter third number: "))
# Initialize GCD
gcd = 1
# Loop to find GCD
for i in range(1, min(num1, num2, num3)+1):
  if(num1%i == 0 and num2%i == 0 and num3%i == 0):
    gcd = i
print("GCD of three numbers is", gcd)
```

OUTPUT:



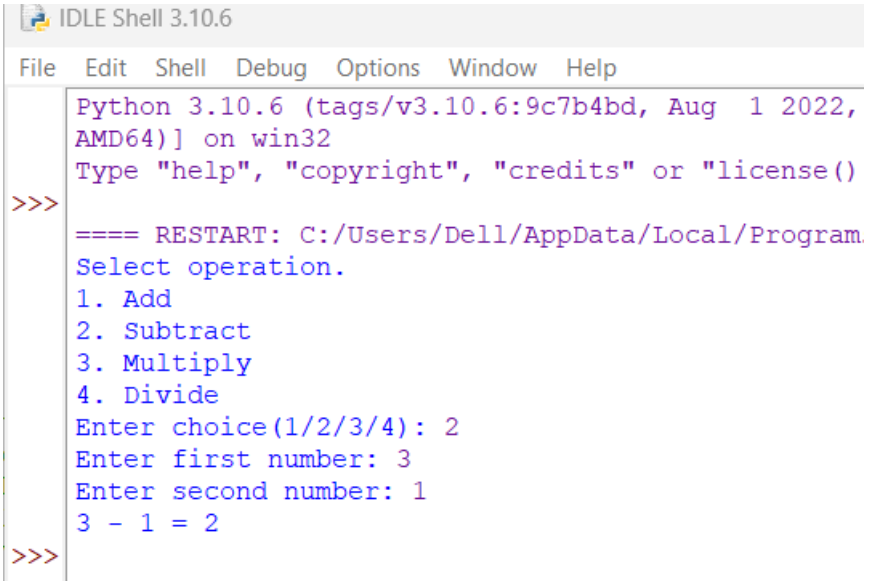
Ex. No:	38	simple calculator program in python
Date:		

PROGRAM:

```
#calculator
import math


def sum(num1, num2):
  sum = num1 + num2
  return sum


def sub(num1, num2):
  sub = num1 - num2
  return sub


def pro(num1, num2):
  product = num1 * num2
  return product


def div(num1, num2):
  quo = num1 / num2
  return quo


#main
ans = 'y'
while ans == 'y':
print('''THIS IS A
SIMPLE CALCULATOR
   1. ADD
   2. SUBTRACTION
   3. MULTIPLY
   4. DIVISION''')
  n = int(input("Enter your
choice:"))
  if n == 1:
    n1 = int(input("Enter
the first number:"))
    n2 = int(input("Enter
the second number:"))
    print("The sum of the
given numbers is:",
sum(n1, n2))
  elif n == 2:
    n1 = int(input("Enter
the first number:"))
    n2 = int(input("Enter
the second number:"))
```

```
    print("The difference of
the given numbers is:",
sub(n1, n2))
  elif n == 3:
    n1 = int(input("Enter
the first number is:"))
     n2 = int(input("Enter
the second number is:"))
     print("The product of
the given number is:",
pro(n1, n2))
  elif n == 4:
    n1 = int(input("Enter
the first number:"))
    n2 = int(input("Enter th
second number:"))
    print("The quotient of
the given number is:",
div(n1, n2))
  else:
    print("INVALID
INPUT!!")
  ans = input("Do you
want to continue ?
(Y/N)...")
```

OUTPUT:



Ex. No:	39	find factors of a number in python
Date:		

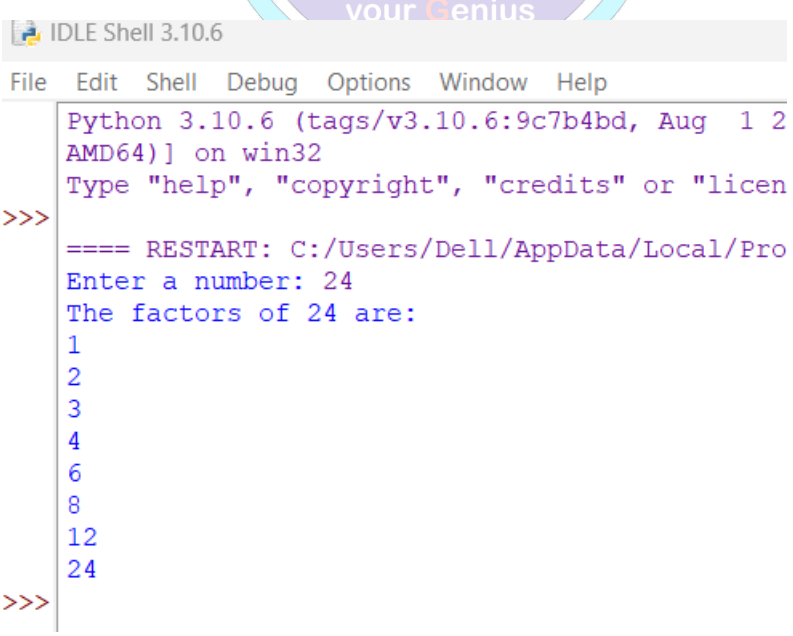
ALGORITHM:

Step1:Start
Step2:Get the input from the user and store in the num
Step3:Compute the loop of num and store the value in i
Step4:And check the condition if the input modulus of i is equal to 0
Step5:print i
Step6:Stop

PROGRAM:

```
num=int(input("Enter a number:"))
print("The factor of ",num,"are")
for i in range(1,num+1):
if num%i==0:
print(i)
```


OUTPUT:



Ex. No:	40	find ASCII value of character in python
Date:		

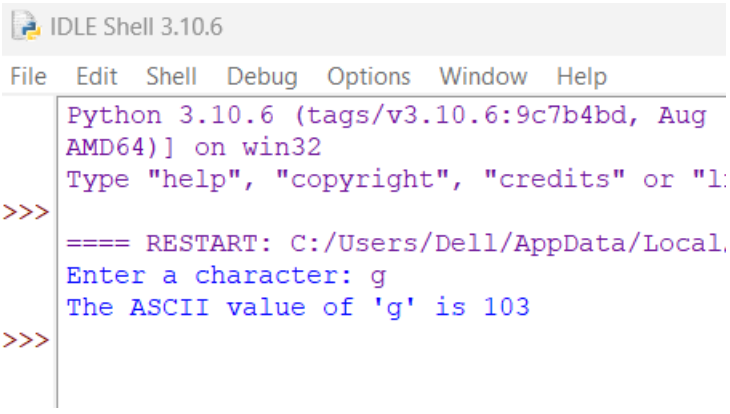
ALGORITHM:

Step1:Start
Step2:Get the input from the user to find the ASCII character
Step3:It can be initialize by using the ord method
Step4:Stop

PROGRAM:

```
char = input("Enter a character: ")

print("The ASCII value of '" + char + "' is", ord(char))
```

OUTPUT:



Ex. No:	41	program to add two numbers
Date:		

ALGORITHM:

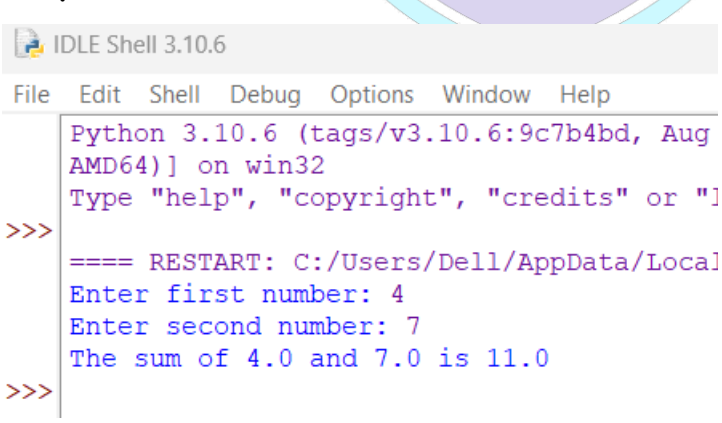
Step1:Start
Step2:Get the input from the user and store in number1
Step3:Get the input from the user and store in number2
Step4:And with the third variable add two numbers with the symbol +
Step5:Stop

PROGRAM CODE:

```
number1 = float(input("Enter first number: "))
number2 = float(input("Enter second number: "))
sum = number1 + number2
print("The sum of {0} and {1} is {2}".format(number1, number2, sum))
```

OUTPUT:

.



Ex. No:	42	program to concatenate 2 lists
Date:		

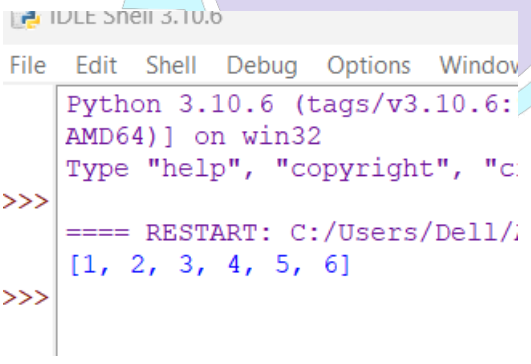
ALGRITHM:

Step1:Start
Step2:Get the value for the list and store in list1
Step3:Get the value for the list and store in list2
Step4:And with the third variable concatenate two list and print the list
Step5:Stop

PROGRAM:

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list3 = list1 + list2
print(list3)
```

OUTPUT:

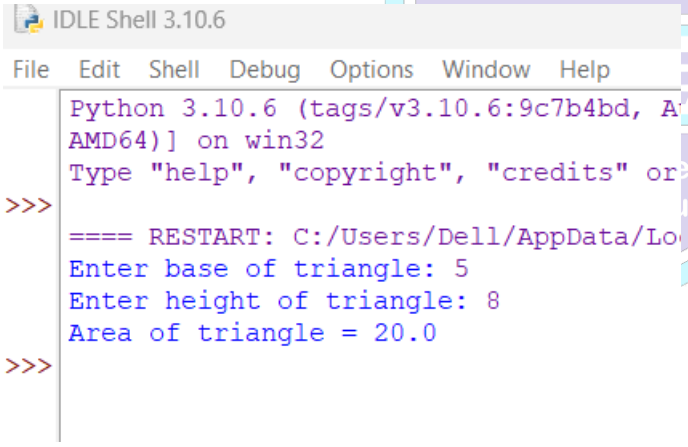


Ex. No:	43	program to calculate area of triangle
Date:		

PROGRAM:

```
# Take inputs
base = float(input("Enter base of triangle: "))
height = float(input("Enter height of triangle: "))

# Calculate area
area = 0.5 * base * height

# Print result
print("Area of triangle =", area)
```

OUTPUT:

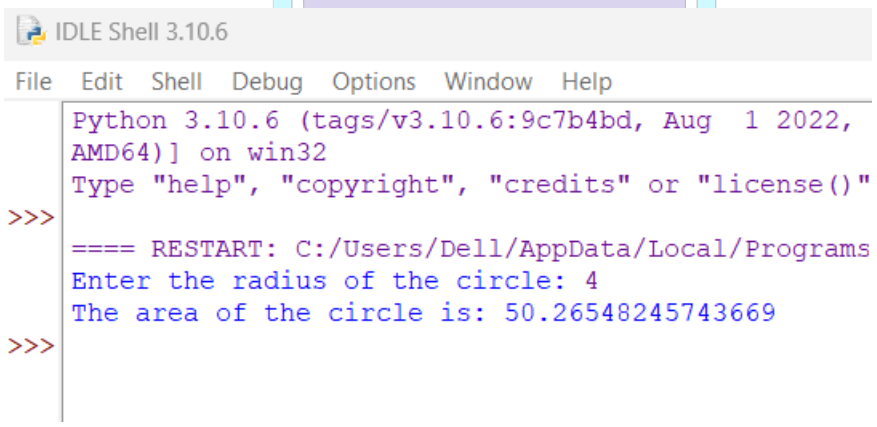


Ex. No:	44	program to calculate area of the circle
Date:		

PROGRAM:

```
# Import math library
import math

# Take radius input from user
radius = float(input('Enter the radius of the circle: '))

# Calculate area
area = math.pi * radius * radius

# Print result
print("The area of the circle is:", area)
```

OUTPUT:

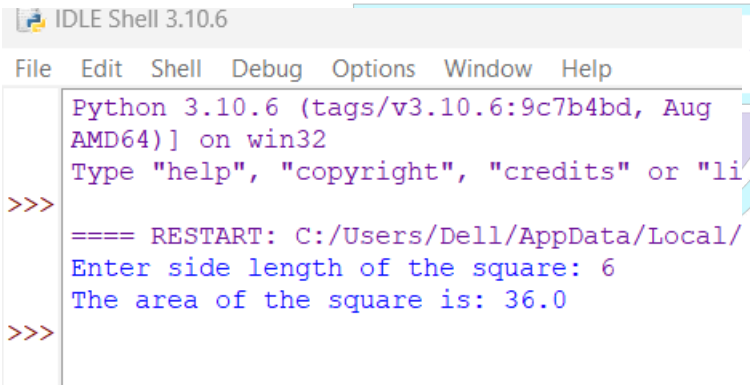


Ex. No:	45	program to calculate area of the square
Date:		

PROGRAM:

```
# Take side length input from user
side = float(input('Enter side length of the square: '))

# Calculate area
area = side * side

# Print result
print('The area of the square is:', area)
```

OUTPUT:

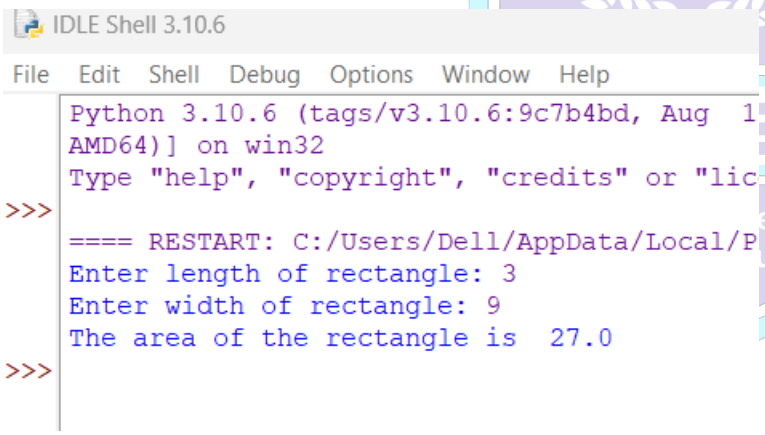


Ex. No:	46	program to calculate area of the rectangle
Date:		

PROGRAM:

```
length = float(input('Enter length of rectangle: '))
width = float(input('Enter width of rectangle: '))
area = length * width
print('The area of the rectangle is ', area)
```

OUTPUT:

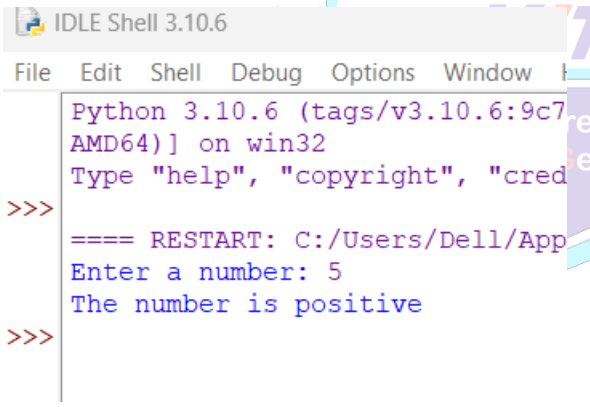


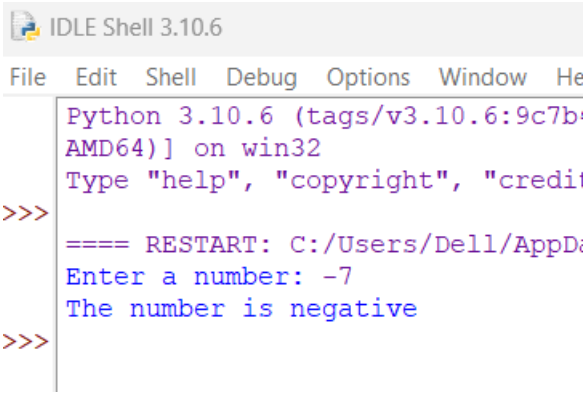
Ex. No:	47	program to check positive or negative number
Date:		

PROGRAM:

```
num = float(input("Enter a number: "))
if num > 0:
   print("The number is positive
elif num == 0:
   print("The number is zero")
else:
   print("The number is negative")
```

OUTPUT:



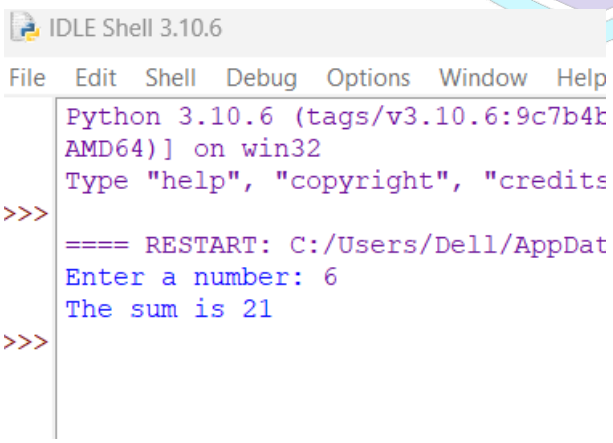


Ex. No:	48	program to find sum of the natural numbers
Date:		

PROGRAM:

```
# Take input from user
num = int(input("Enter a number: "))
sum = 0
# iterate until num
for i in range(1, num+1):
    sum += i
print("The sum is", sum)
```

OUTPUT:



Ex. No:	49	program to display the day of entered date
Date:		

PROGRAM:

```
import datetime

date_str = input("Enter a date (DD/MM/YYYY): ")
date_obj = datetime.datetime.strptime(date_str, "%d/%m/%Y")
print("The day is:", date_obj.strftime("%A"))
```

OUTPUT:

Ex. No:	50	program to count vowels in a entered string
Date:		

ALGORIHTM:

Step1:Start

Step2:Get the input from the user

Step3:Take the value of vowels as 0

Step4:Compute the loop of string and store the value in char

Step1:check the condition if input exists in any vowels characters

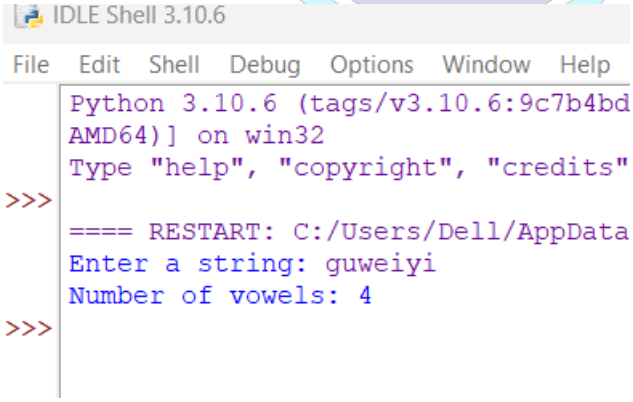
Step1:then calculate the number of vowels in the input and finally print vowels

Step1: Stop

PROGRAM:

```
string = input("Enter a string: ")
        vowels = 0
        for char in string:
           if char in 'aeiouAEIOU':
              vowels = vowels + 1

        print("Number of vowels:", vowels)
```

OUTPUT:



Ex. No:	51	program to slice the list values
Date:		

ALGORITHM:

Step1:Start

Step1:Number of values in the list and store in the variable numbers

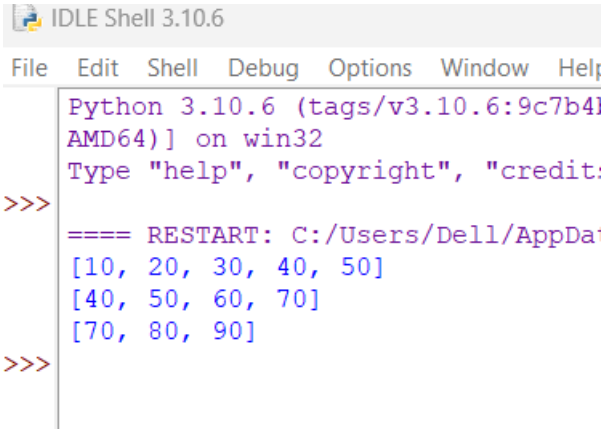
Step1:and do the following slicing operations

Step1:Stop

PROGRAM:

```
numbers = [10, 20, 30, 40, 50, 60, 70, 80, 90]
print(numbers[:5])
print(numbers[3:7])
print(numbers[-3:])
```


OUTPUT:



Ex. No:	52	program to covert bytes into a string
Date:		

ALGORITHM:

Step1:Start

Step1:Get the value for the bytes to convert

Step1:Compute the decode in the string and store in then str_obj

Step1:and print the str_obj

Step1:Stop

PROGRAM:

```
bytes_obj = b'Hello World'
str_obj = bytes_obj.decode()

print(str_obj)
```

OUTPUT:

Ex. No:	53	program to count number of occurence of a particular character in a list or string
Date:		

ALGORITHM:

Step1:Start

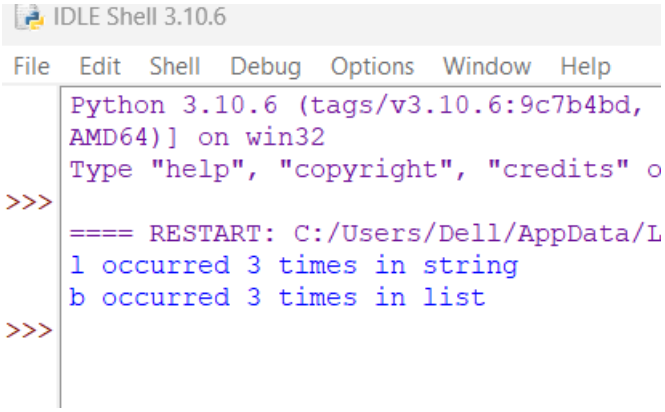
Step1:

PROGRAM:

:

```
# For string
string = "Hello World"
char = 'l'
count = 0
for c in string:
  if c == char:
     count += 1
     print(f"{char} occurred{count} times in string")

# For list my_list =
my_list= ['a','b','c','b','a','b']
char = 'b'
count = 0
for item in my_list:
  if item == char:
      count += 1
print(f"{char} occurred{count} times in list")
```

OUTPUT:



Ex. No:	54	program to create and access elements in a dictionaryv
Date:		

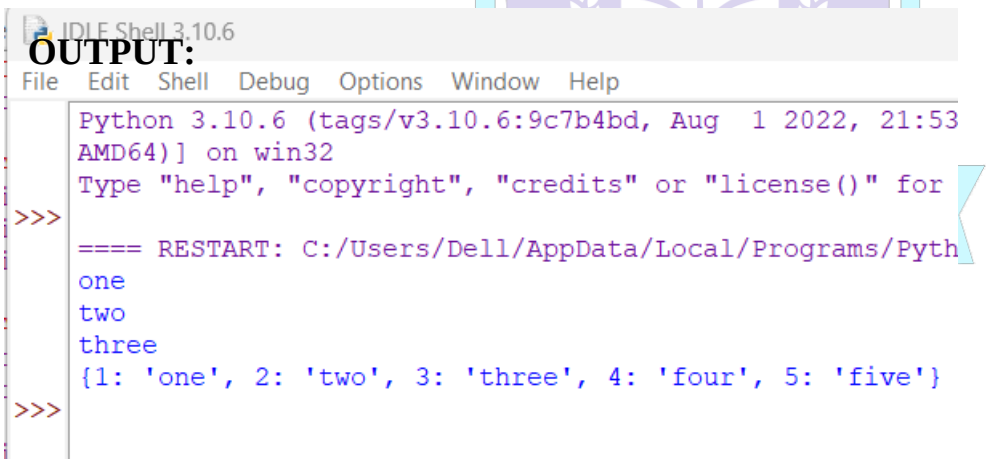
ALGORITHM:

Step1:Start

PROGRAM:

```
dict = {1:'one', 2:'two', 3:'three'}
print(dict[1])
print(dict[2])
print(dict[3])
dict[4] = 'four'
dict[5] = 'five'
print(dict)
```

OUTPUT:



Ex. No:	55	program to check magic number
Date:		

ALGORITHM:

Import the math module to use the log10 function.

Prompt the user to enter a number.

Calculate the number of digits in the input number using digitCount = int(math.log10(num)) + 1.

Initialize sumOfDigits and temp variables to 0 and the input number, respectively.

Enter a while loop that continues until digitCount becomes 1.

Within the loop, reset sumOfDigits to 0.

Enter another while loop that calculates the sum of the digits of temp.

Update temp by dividing it by 10 (integer division).

Update sumOfDigits with the sum of the digits.

Calculate the number of digits in sumOfDigits and update digitCount.

After exiting the while loop, check if sumOfDigits is equal to 1.

If it is, print "Magic number".

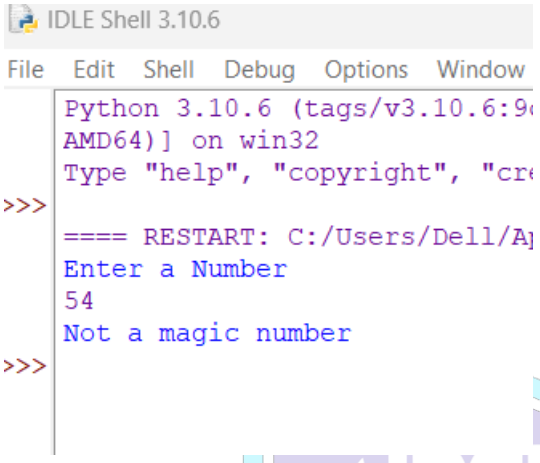
If it's not, print "Not a magic number".

PROGRAM:

```
import math
num = int(input("Enter a Number \n"))
digitCount = int(math.log10(num))+1
sumOfDigits = 0
temp = num
while( digitCount > 1):
    sumOfDigits = 0
    while(temp > 0):
        sumOfDigits +=temp%10
        temp = temp//10
    temp = sumOfDigits
    digitCount =int(math.log10(sumOfDigits))+1
if(sumOfDigits == 1):
    print("Magicnumber")
else:
```

```
print("Not a magic number")
```

OUTPUT:



Ex. No:	56	program to check armstrong number
Date:		

PROGRAM:

```
num = 1634
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** 4
    temp //= 10
if num == sum:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")
```

OUTPUT:

Ex. No:	57	program to find the length of the list
Date:		

PROGRAM:

```
# Declare a list
my_list = [1, 3, 5, 7, 9]

# Get length of list
list_len = len(my_list)

print("The length of the list is:", list_len)
```

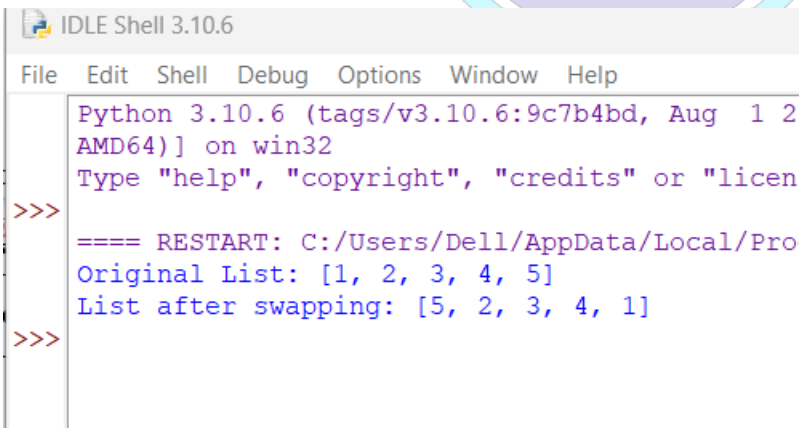
OUTPUT:

Ex. No:	58	swap first and last elements of a list
Date:		

PROGRAM:

```
def swap_first_last(list):
  size = len(list)
  # Check if list is empty
  if not size:
    return list
  # Save first and last elements
  first = list[0]
  last = list[size - 1]
  # Swap elements
  list[0] = last
  list[size - 1] = first
  return list
list_a = [1, 2, 3, 4, 5]
print("Original List:", list_a)
list_a  =  swap_first_last(list_a)
print("List after swapping:", list_a)
```

OUTPUT:

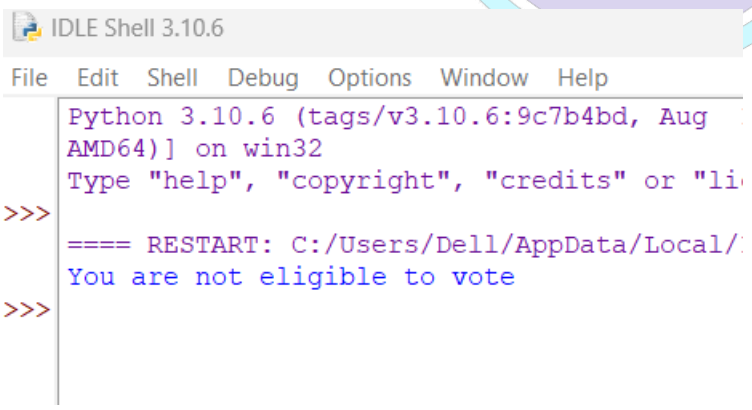


Ex. No:	59	program to determine voting eligibility
Date:		

PROGRAM:

```
age = 17

if age >= 18:
  print("You are eligible to vote")
else:
  print("You are not eligible to vote")
```

OUTPUT:



Ex. No:	60	program to calculate mark
Date:		

PROGRAM:

```
sub1 = 85
sub2 = 95
sub3 = 88

max_marks = 100

total_marks = sub1 + sub2 + sub3
print("Total  Marks obtained:",  total_marks)

percentage = (total_marks/300) * 100
print("Percentage  Marks:  ",  percentage)

if percentage >= 90:
    grade = 'A'
elif percentage >= 80:
    grade = 'B'
elif percentage >= 60:
    grade = 'C'
else:
    grade = 'D'

print("Grade:", grade)
```

OUTPUT:

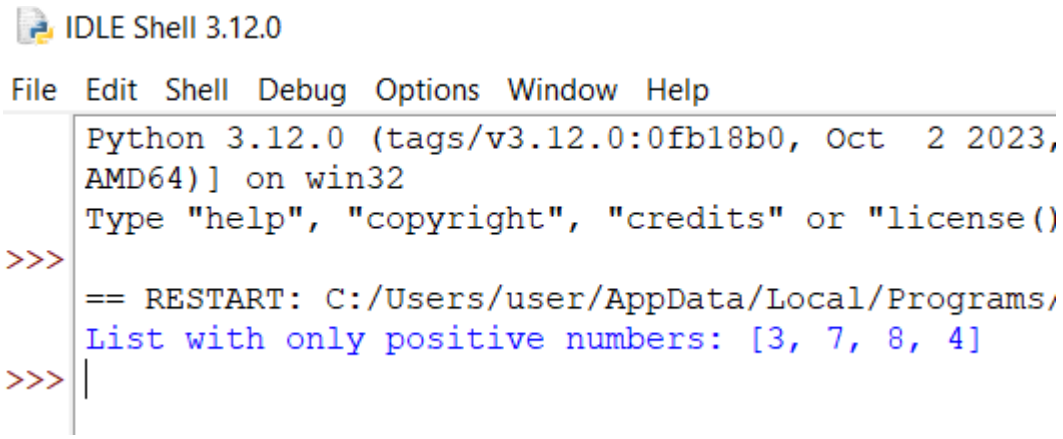
```
IDLE Shell 3.10.6
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022,
AMD64)] on win32
Type "help", "copyright", "credits" or "license(
>>>
==== RESTART: C:/Users/Dell/AppData/Local/Prograı
Total Marks obtained: 268
Percentage Marks:  89.33333333333333
Grade: B
>>>
```

Ex. No:	61	ADD ONLY POSITIVE NUMBER IN A LIST
Date:		

PROGRAM:

```
lst = [1,2,3,-4,6,-7,-10]
for i in lst:
  if i>=0:
    x = i
    print("The positive number in a list are",x
```

OUTPUT:

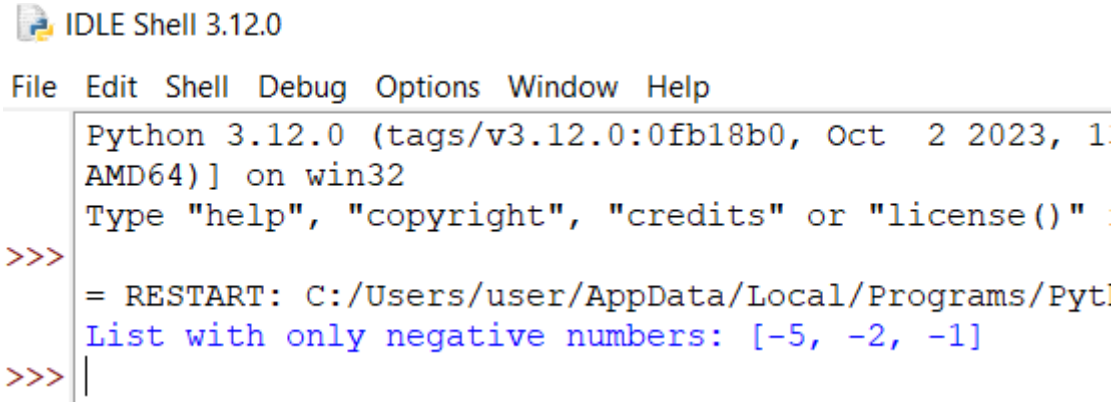


Ex. No:	62	ADD ONLY NEGATIVE NUMBER IN A LIST
Date:		

PROGRAM:

```
lst = [1,2,3,-4,6,-7,-10]
for i in lst:
  if i<=0:
    x = i
    print("The negative number in a list are",x)
```

OUTPUT:

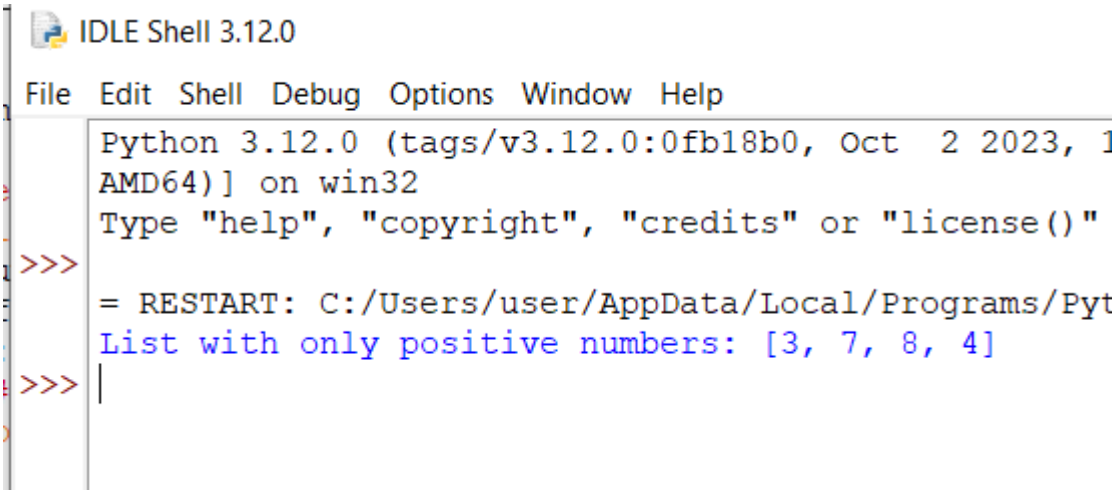


Ex. No:	63	USING IF AND ELSE
Date:		

PROGRAM:

```
age = int(input("Enter your age:"))
if age>=18:
  print("you are eligible for voting")
else:
  print("you are not eligible for voting")
```

OUTPUT:

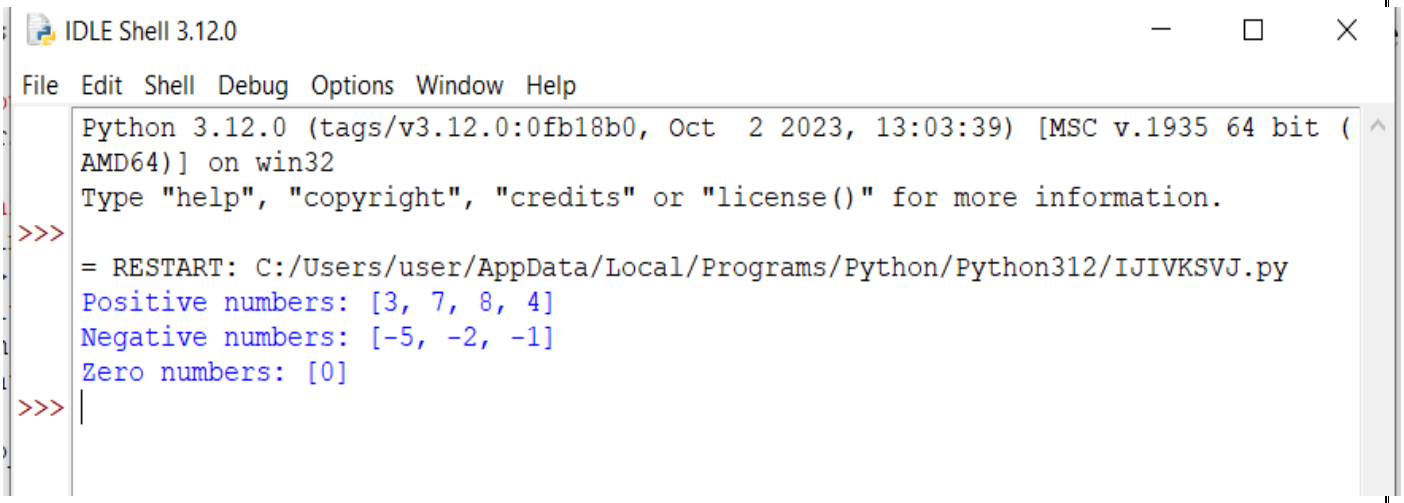


Ex. No:	64	USING IF,ELIF,ELSE
Date:		

PROGRAM:

```
age = int(input("Enter your age:"))
if age>18:
  print("you are eligible for voting")
elif age==18:
  print("you are eligible for voting")
else:
  print("you are not eligible for voting")
```

OUTPUT:

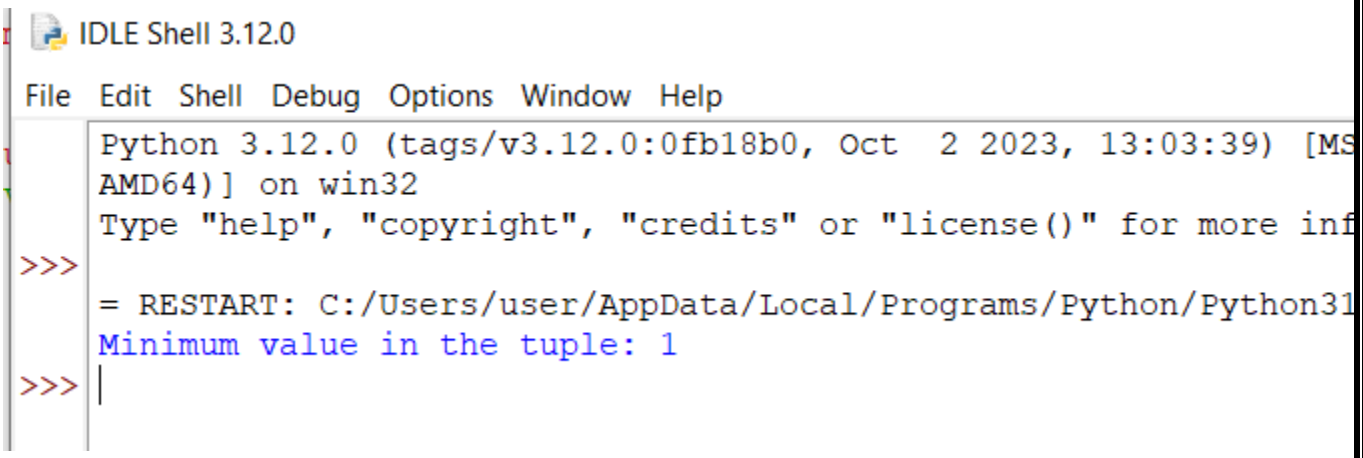


Ex. No:	65	MINIMUM VALUE IN A TUPLE
Date:		

PROGRAM:

```
tup = (90,42,54,34,18)
print("The minimum value in the tuple is",min(tup))
```

OUTPUT:

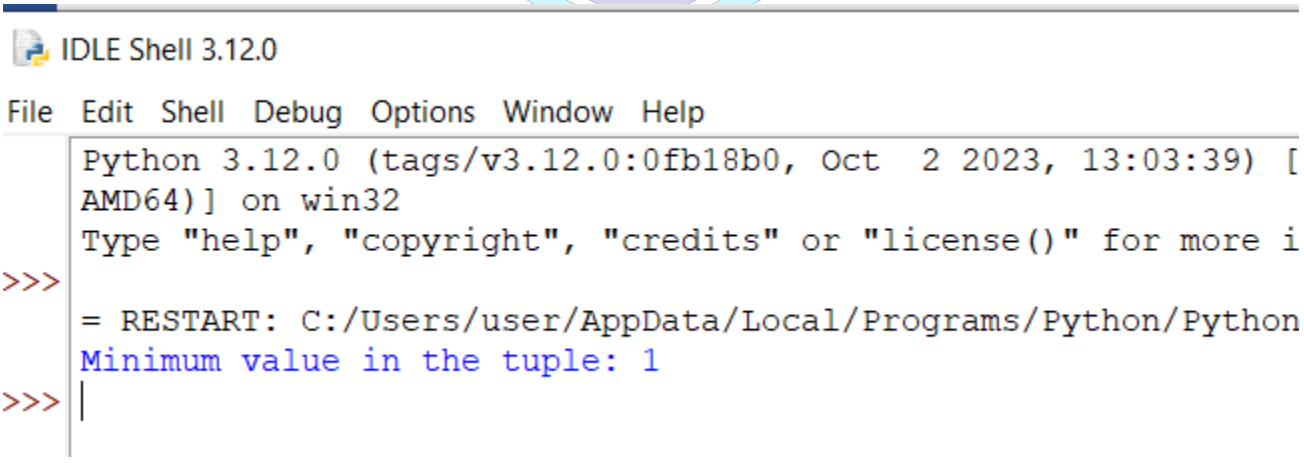


Ex. No:	66	MAXIMUM VALUE IN A TUPLE
Date:		

PROGRAM:

```
# Sample tuple
my_tuple = (5, 3, 8, 2, 1, 9, 4)

# Initialize the minimum value with the first element of the tuple
min_value = my_tuple[0]

# Iterate through the tuple to find the minimum value
for num in my_tuple:
    if num < min_value:
        min_value = num

# Print the result
print("Minimum value in the tuple:", min_value)
```

OUTPUT:

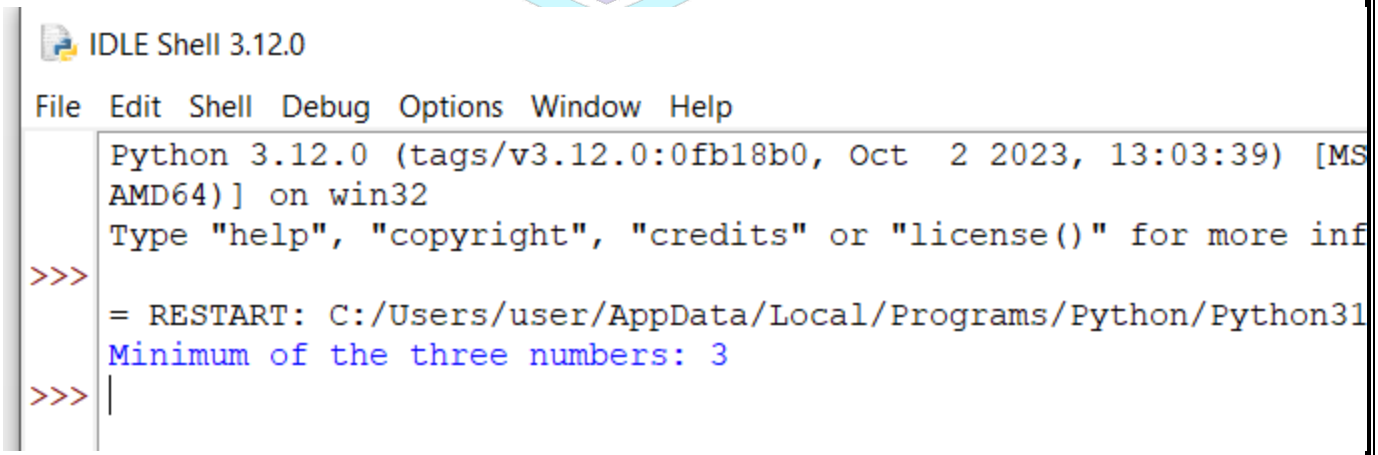


Ex. No:	67	MINIMUM OF THREE NUMBERS
Date:		

PROGRAM:

```
# Three numbers
num1 = 5
num2 = 8
num3 = 3

# Using if and else statements
if num1 <= num2 and num1 <= num3:
    minimum = num1
elif num2 <= num1 and num2 <= num3:
    minimum = num2
else:
    minimum = num3

# Print the result
print("Minimum of the three numbers:", minimum)
```

OUTPUT:

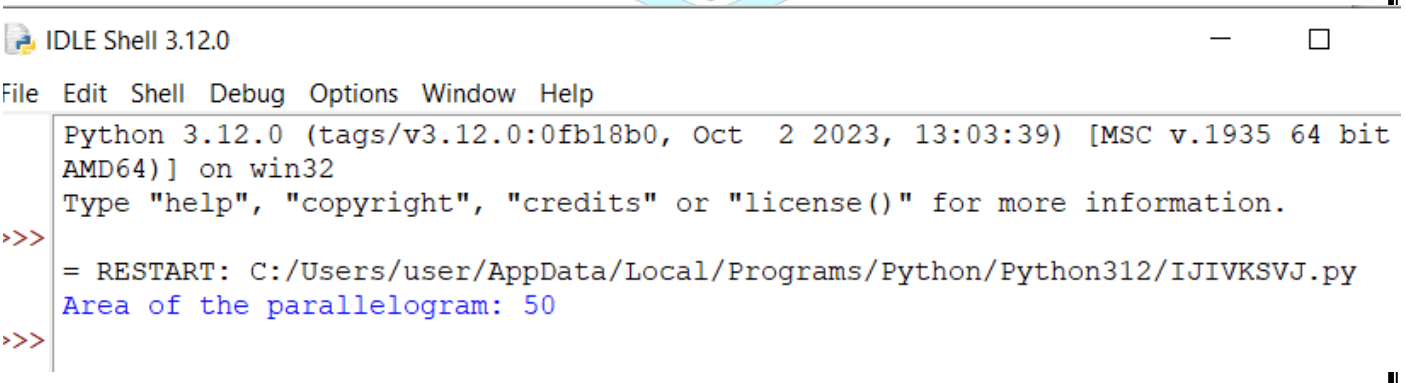


Ex. No:	68	AREA OF PARELLOGRAM
Date:		

PROGRAM:

```
# Function to calculate the area of a parallelogram
def parallelogram_area(base, height):
    area = base * height
    return area

# Example values for base and height
base_length = 10
height = 5

# Calculate the area of the parallelogram
area_of_parallelogram = parallelogram_area(base_length, height)

# Print the result
print("Area of the parallelogram:", area_of_parallelogram)
```

OUTPUT:

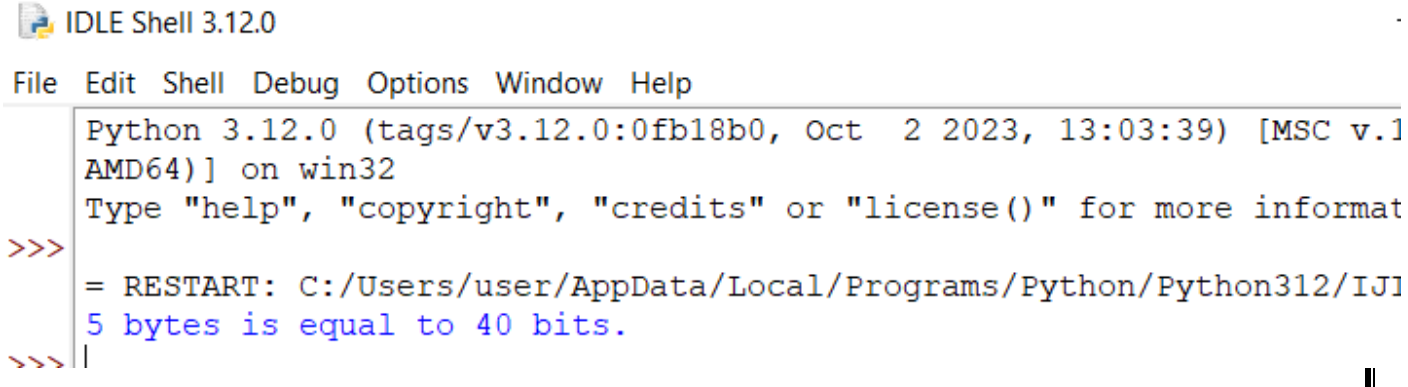


Ex. No:	69	CONVERT BYTES INTO BITS
Date:		

PROGRAM:

```
# Function to convert bytes to bits
def bytes_to_bits(bytes_value):
    bits_value = bytes_value * 8
    return bits_value

# Example value in bytes
bytes_value = 5

# Convert bytes to bits
bits_value = bytes_to_bits(bytes_value)

# Print the result
print(f"{bytes_value} bytes is equal to {bits_value} bits.")
```


OUTPUT:



Ex. No:	70	PERFORMING DICTIONARY OPERATIONS
Date:		

PROGRAM:

Creating a Dictionary:

```
my_dict = {"name": "John", "age": 25, "city": "New York"}
```

Accessing Values:

```
print("Name:", my_dict["name"])
print("Age:", my_dict["age"])
```

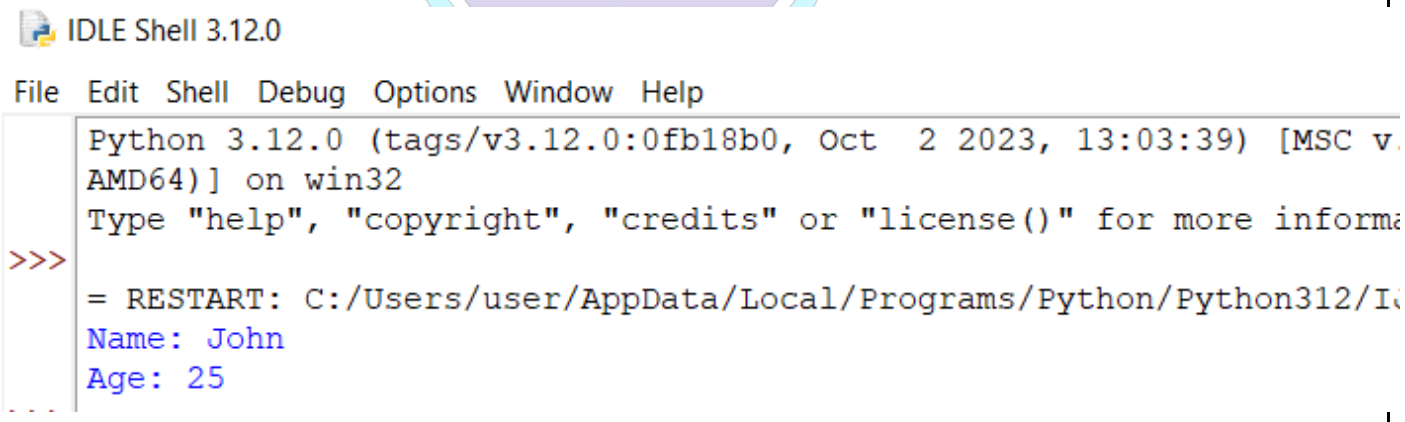
Adding or Updating a Key-Value Pair:

```
my_dict["occupation"] = "Engineer"
my_dict["age"] = 26  # Updating the value of an existing key
```

Removing a Key-Value Pair:

```
del my_dict["age"]
```

OUTPUT:

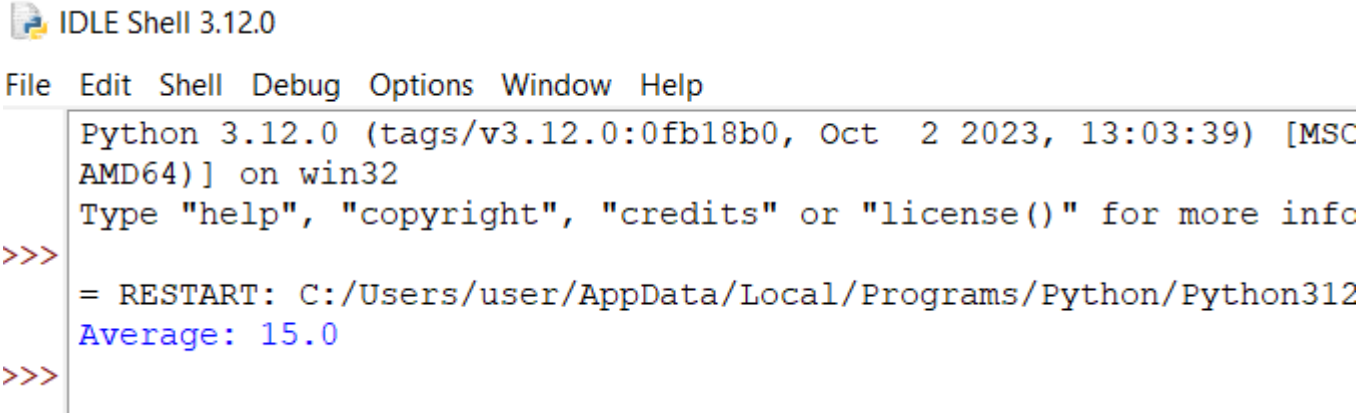


Ex. No:	71	AVERAGE OF ELEMENTS IN A LIST
Date:		

PROGRAM:

```
# Function to calculate the average of elements in a list
def calculate_average(numbers):
    if not numbers:
        return None  # Return None for an empty list to avoid division by zero

    # Calculate the average
    average = sum(numbers) / len(numbers)
    return average

# Example list of numbers
number_list = [5, 10, 15, 20, 25]

# Calculate the average of the list
average_value = calculate_average(number_list)

# Print the result
print("Average:", average_value)
```

OUTPUT:



Ex. No:	72	CHECK IF A NUMBER IS PALINDROME OR NOT
Date:		

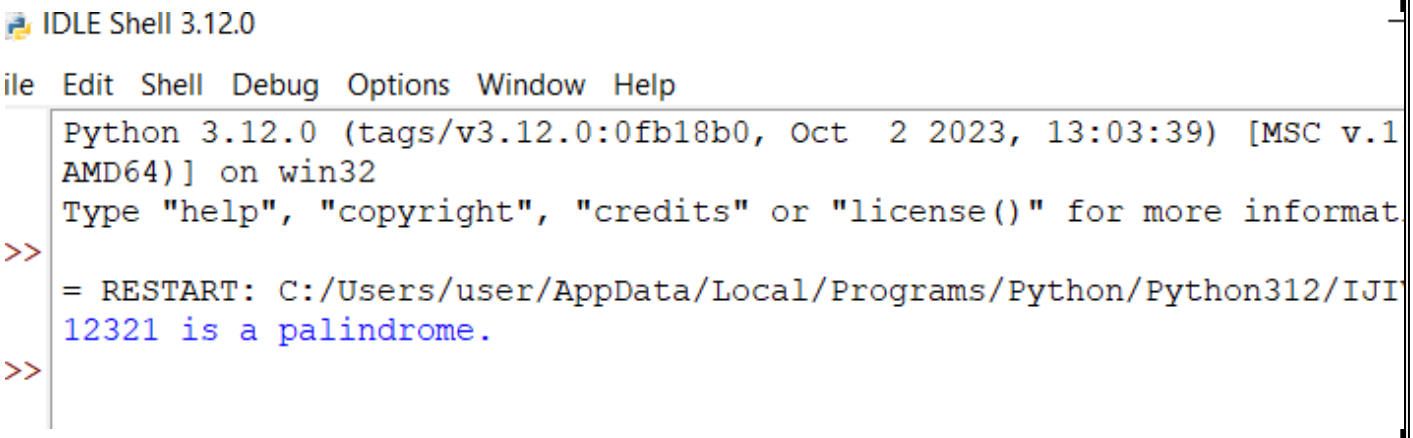
ALOGORITHM:

It creates a set of all the consonant letters in the English alphabet.
It counts the number of consonants in the input string using a generator expression and the sum() function.
It returns the count of the consonants.
It assigns the input string "Hello World" to the variable input_string
It calls the count_consonants function with the input_string, storing the result in a variable called result.
Lastly, it prints the number of consonants in the input string along with the actual count.

PROGRAM:

```
def checkPalindrome(str):
    for i in range(0, len(str)//2):
        if str[i] != str[len(str)-i-1]:
            return False
    return True
st = input("Enter the input")
if(checkPalindrome(st) == True):
    print(st,"is a palindrome")
else:
    print(st,"is not a palindrome"
```

OUTPUT:



Ex. No:	73	COUNT THE NUMBER OF CONSTANTS IN A STRING
Date:		

ALGORITHM:

The function count_consonants takes an input string as a parameter.

It initializes a set called consonants that contains all the consonant letters in both lowercase and uppercase.

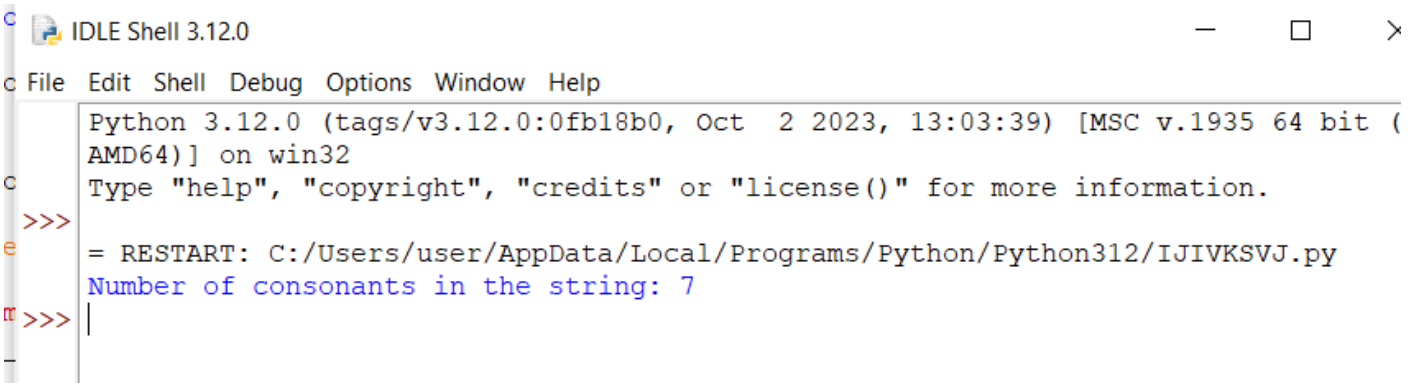
It uses a generator expression and the sum function to count the number of characters in the input string that are present in the consonants set.

Finally, it returns the count of consonants.

PROGRAM:

```
def count_consonants(input_string):
consonants = set("bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ")
consonant_count = sum(1 for char in input_string if char in consonants)
return consonant_count

input_string = "Hello World"
result = count_consonants(input_string)
print("Number of consonants in the string:", result)
```

OUTPUT:



Ex. No:	74	FIND THE INTERSECTION OF TWO LISTS
Date:		

ALGORITHM :

Step1:Start

Step1:Get the input from the user for both list1 and list2

Step1:And by using the intersection function set intersection in the variable

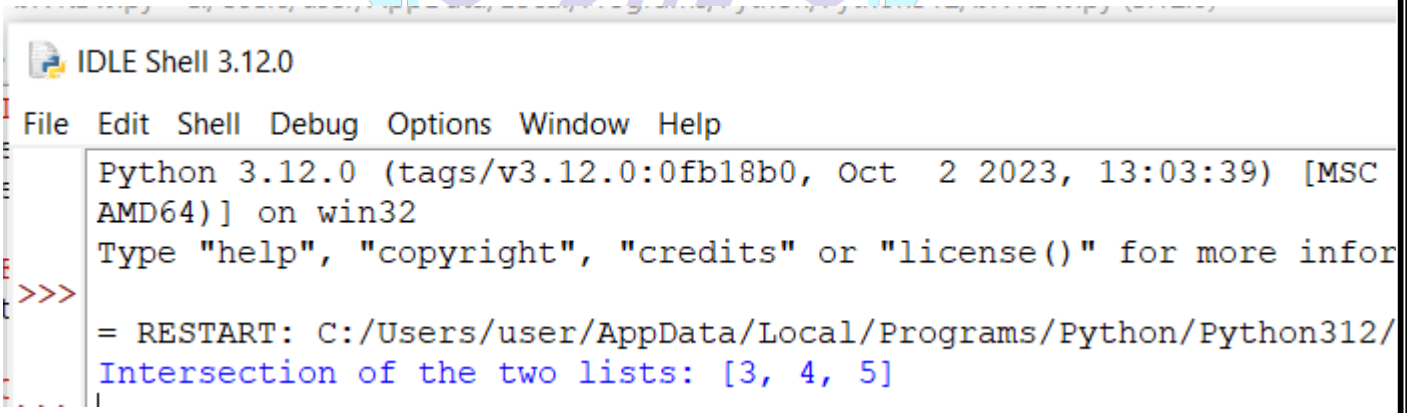
Step1:print intersection_list

Step1:Stop

PROGRAM:

```
list1 = [1, 2, 3, 4, 5]
list2 = [3, 4, 5, 6, 7]
intersection_set = set(list1).intersection(list2)
intersection_list = list(intersection_set)
print("Intersection of the two lists:", intersection_list)
```

OUTPUT:



Ex. No:	75	POWER OF N NUMBERS
Date:		

ALGORITHM:

Step1:start

Step1:define the function base and experiment

Step1:And by using the double exponenet form the equation

Step1:And outside the function get the input for both base and exponent

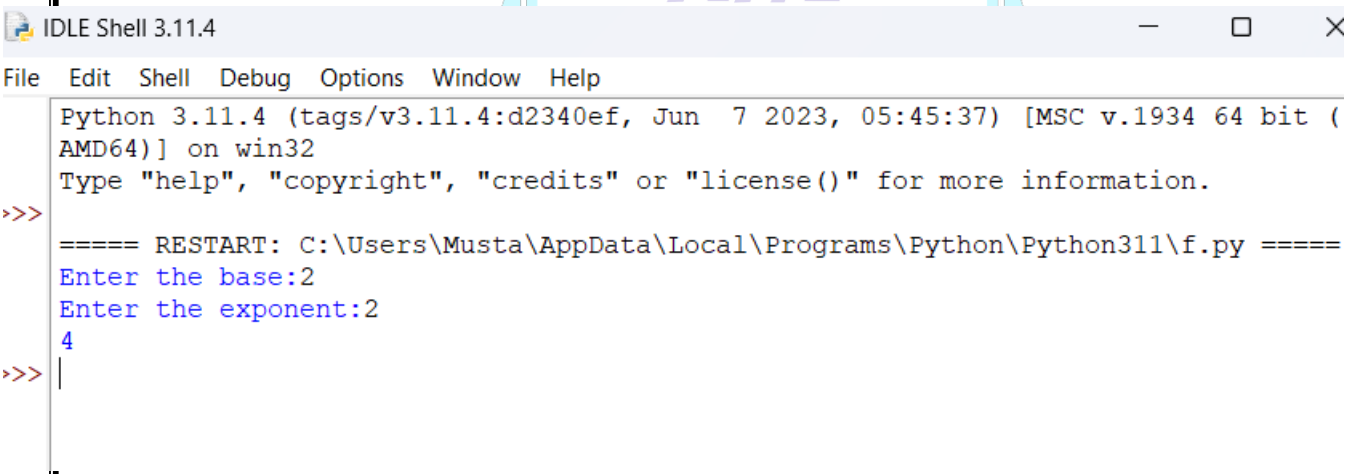
Step1:And finally call the function

Step1:Stop

PROGRAM:

```
def power_of_number(base, exponent):
  return base ** exponent
base = int(input("Enter the base:"))
exponent = int(input("Enter the exponent:"))
print(power_of_number(base, exponent))
```

OUTPUT:



Ex. No:	76	Python program to find the second largest number in a list.
Date:		

ALGORITHM:

Step1:Start
Step1:Get the value of the list and store it in list1
Step1:And In the other variable list2 set list1
Step1:And then sort the list1 and store it in list2
Step1:print list2
Step1:Stop

PROGRAM:

```
list1 = [10,, 20, 4, 41, 45, 99]
list2 = list(set(list1))
list2.sort()
print("Second largest element is:", list2[-2]))
```

OUTPUT:

```
>>
===== RESTART: C:\Users\Musta\AppData\Local\Programs\Python\Python311\m.py =====
Second largest element is: 45
>>
```

Ex. No:	77	ython program to find the second smallest number in a list.
Date:		

ALGORITHM:

Step1:Start

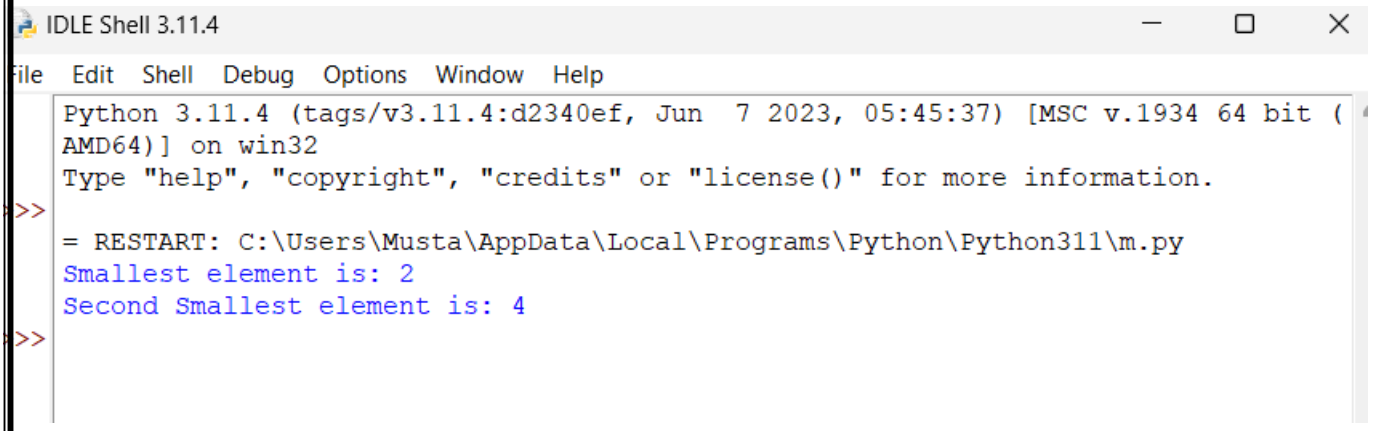
Step1:Define the function find_len(list1)

Step1:

PROGRAM:

```
def find_len(list1):
    length = len(list1)
    list1.sort()
    print("Smallest element is:", list1[0])
    print("Second Smallest element is:", list1[1])
list1=[12, 45, 2, 41, 31, 10, 8, 6, 4]
Largest = find_len(list1)
```

OUTPUT:



Ex. No:	78	Python program to check if a number is a perfect square
Date:		

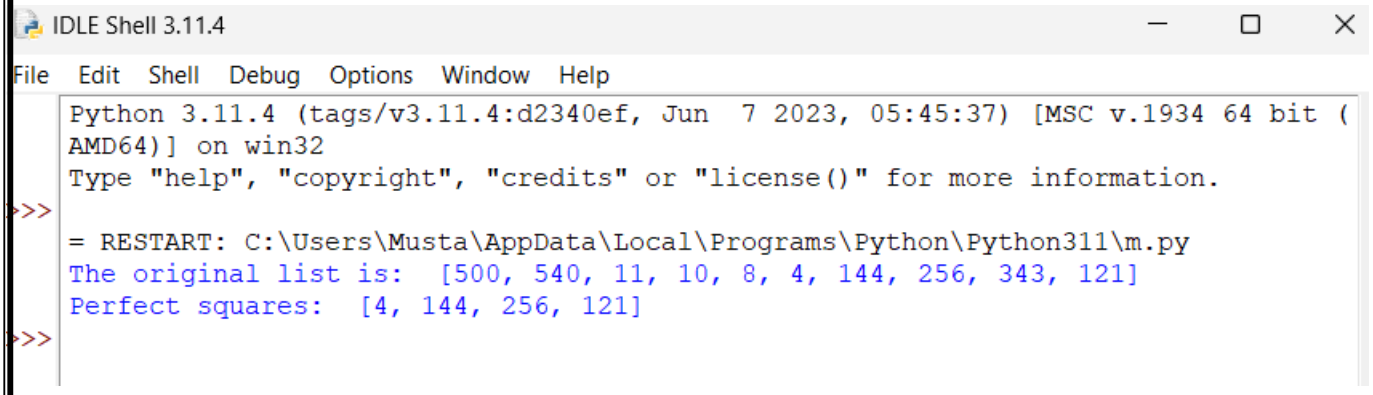
ALGORITHM:

Step1:Start
Step1:import the math module
Step1:Get the input from the user to check the perfect square
Step1:Define a list called Nums that contains the numbers you want to check for perfect squares.
Step5: Use list comprehension to create a new list called p_square that contains the elements from Nums for which the square root of each element is equal to the floor of the square root
Step5:Stop

PROGRAM:

```
import math
Nums = [500, 540, 11, 10, 8, 4, 144, 256, 343, 121]
print("The original list is: ", Nums)
p_square = [i for i in Nums if (math.sqrt(i) == math.floor(math.sqrt(i)))]
print("Perfect squares: ", p_square)
```

OUTPUT:



Ex. No:	79	Program to calculate the square root of a number without using the built-in sqrt function
Date:		

ALGORITHM:

The findSqrt function takes a number x as input.

It first checks if the number is less than 2. If it is, the function returns the number itself, as the square root of any number less than 2 is the number itself.

If the number is greater than or equal to 2, the function initializes two variables: y and z. y is set to the original number x, and z is calculated as the average of y and x/y.

The function then enters a loop that continues until the absolute difference between y and z is less than 0.00001.

In each iteration of the loop, y is updated to the value of z, and z is updated as the average of y and x/y.

Once the loop exits, the function returns the final value of z, which is an approximation of the square root of the original number x.

In the main section of the code (if __name__== '__main__':), a number n is defined as 323242.

The findSqrt function is called with n as the argument, and the result is stored in the variable ans.

Finally, the value of ans is printed.

PROGRAM:

```
def findSqrt(x):
    if x < 2:
        return x
    y = x
    z = (y + (x/y)) / 2
    while abs(y - z) >= 0.00001:
        y = z
        z = (y + (x/y)) / 2
    return z
if__name__== '__main__':
    n = 323242

    ans = findSqrt(n)
    print(ans)
```

OUTPUT:

```
>>>
===== RESTART: C:\Users\Musta\AppData\Local\Programs\Python\Python311\m.py =====
568.543753813196
>>>
```

Ex. No:	80	Python program to check if a given number is a strong number or not.
Date:		

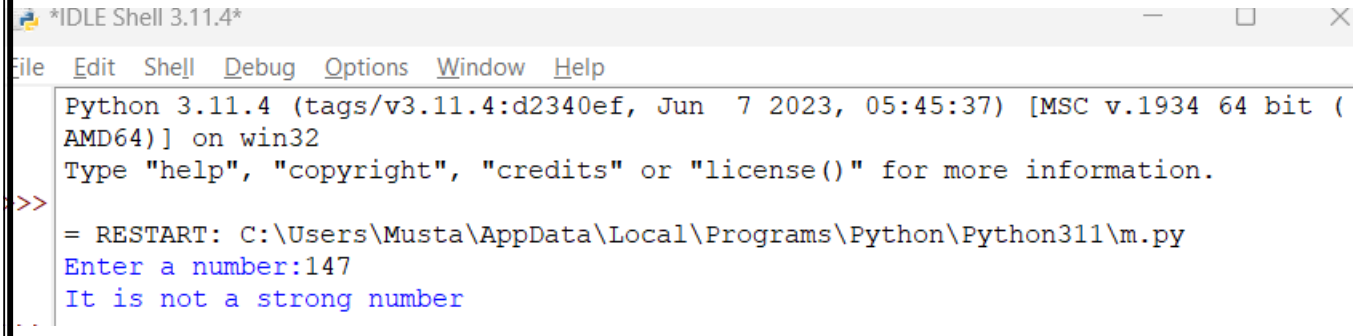
ALGORITHM:

1. The is_strong function takes an integer n as input.
2. It converts the number n into a list of its individual digits using a list comprehension: digits = [int(d) for d in str(n)].
3. It initializes a variable factorial_sum to keep track of the sum of the factorials of the digits.
4. It iterates over each digit in the digits list and calculates the factorial of each digit using a nested loop.
5. The factorial of a digit d is calculated by multiplying all the numbers from 1 to d together.
6. The calculated factorial is added to the factorial_sum.
7. After calculating the factorial sum of all the digits, the function checks if the factorial_sum is equal to the original number n.
8. If the factorial_sum is equal to n, it returns the string "It is a strong number".

PROGRAM:

```
def is_strong(n):
    digits = [int(d) for d in str(n)]
    factorial_sum = 0
    for d in digits:
        f = 1
        for i in range(1, d+1):
            f *= i
        factorial_sum += f
    if factorial_sum == n:
        return "It is a strong number"
    else:
        return "It is not a strong number"
n=int(input("Enter a number:"))
print(is_strong(n))
```

OUTPUT:



Ex. No:	81	Python program to find the union of two lists
Date:		

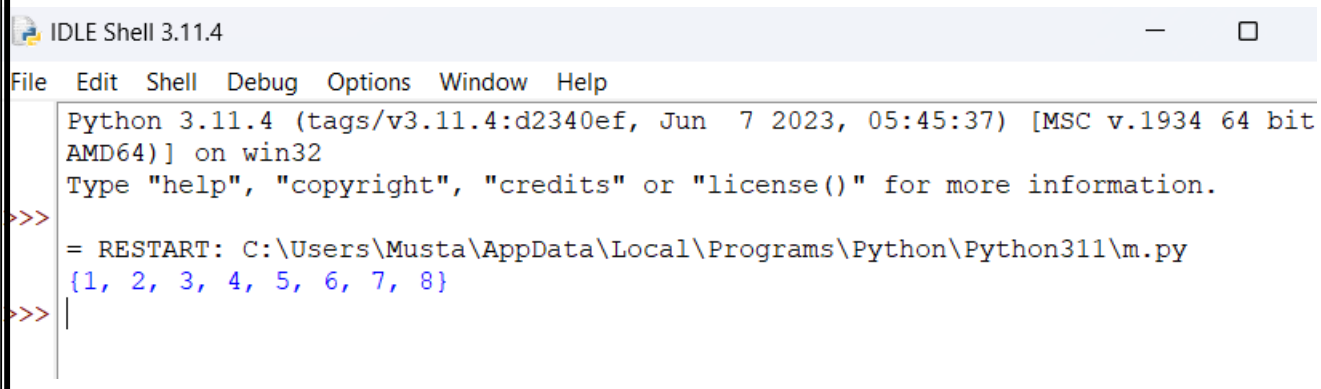
ALGORITHM:

Step1:Start
Step1:Get the input from the user for str_1
Step1:Get the input from the user for str_2
Step1:And union both the str input by using "|" operator
Step1:Stop

PROGRAM:

```
set_1 = {1,2,3,4}
set_2 = {5,6,7,8}
print(set_1|set_2)
```

OUTPUT:



Ex. No:	82	Python program to check if a given email is a valid
Date:		

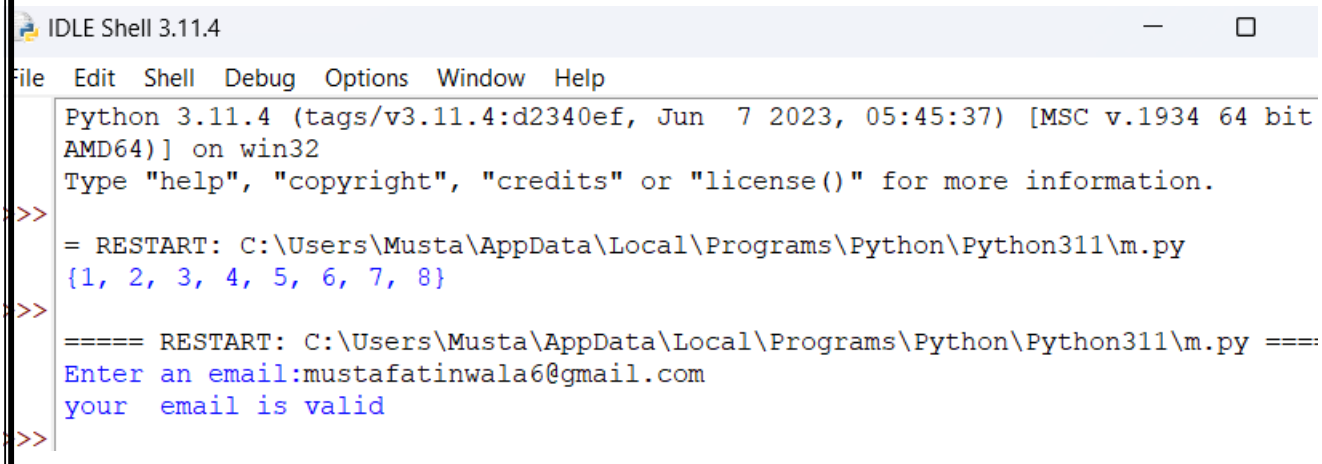
ALGORITHM:

Step1:Start
Step1:Get the input from the user for the email
Step1:check the input using slicing whether it contains .com or not
Step1: check for @gmail using slicing In the input
Step1:print your email is valid
Step1:otherwise your email is not valid
Step1:Stop

PROGRAM:

```
str_1 = input("Enter an email:")
if str_1[-1:-5] == ".com":
print("Your email is not valid")
elif str_1[-10:-6] == "@gmail":
print("your email is valid")
else:
  print("your  email is valid")
```

OUTPUT:



83	Python program to check if a given number is a perfect cube

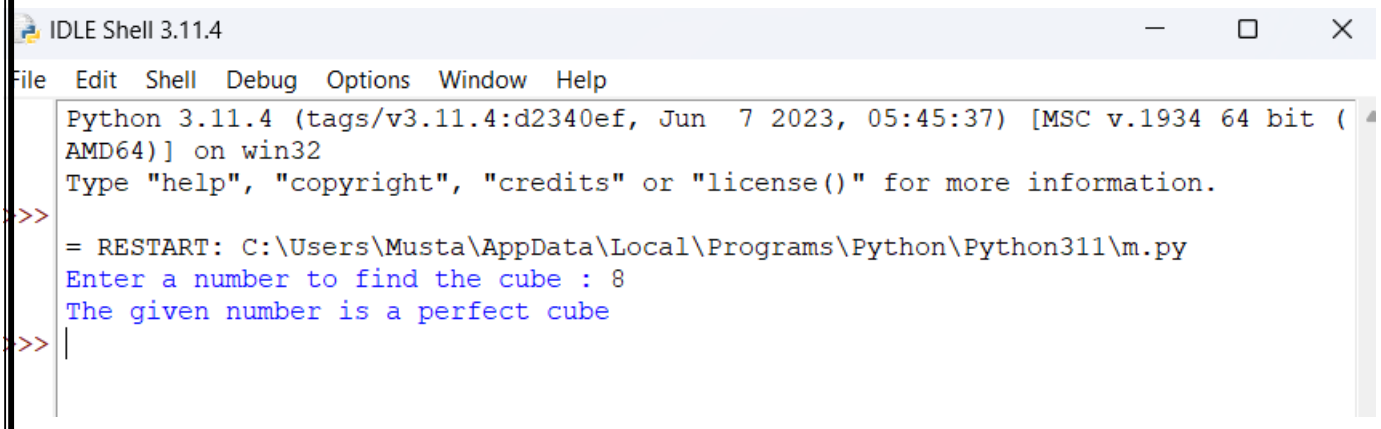
ALGORITHM:

Step1:Start
Step1:Get the input from the user
Step1:Take the other variable named cuberoot check the condition of the root
Step1: round(n(1/3))**3**
Step1:check whether the input is equal to the cube root
Step1:print The given number is perfect cube
Step1:otherwise the given number is not a perfect cube
Step1:Stop

PROGRAM:

```
n = int(input("Enter a number to check the
cube : "))
cube_root =round(n**(1/3))**3
if cube_root == n:
  print("The given number is a perfect
cube")
else:
  print("The given number is not a perfect
cube")
```

OUTPUT:



Ex. No:	84	python program to repetition of tuple
Date:		

ALGORITHM:

Step1:Start
Step1:Take the variable num and store the value 1,2,3
Step1:Get the input from the user and store I repetition
Step1:print num multiplies to repetition
Step1:Stop

PROGRAM:

```
num = (1,2,3)
repetiton  = int(input("Enter a number of
repetitons:"))
print("The repetiton of the tuple
is",num*repetiton)
```

OUTPUT:

```
>>>
===== RESTART: C:\Users\Musta\AppData\Local\Programs\Python\Python311\m.py ====
Enter a number of repetitons:2
The repetiton of the tuple is (1, 2, 3, 1, 2, 3)
>>>
```

Ex. No:	85	Python program to find the area of a regular polygon
Date:		

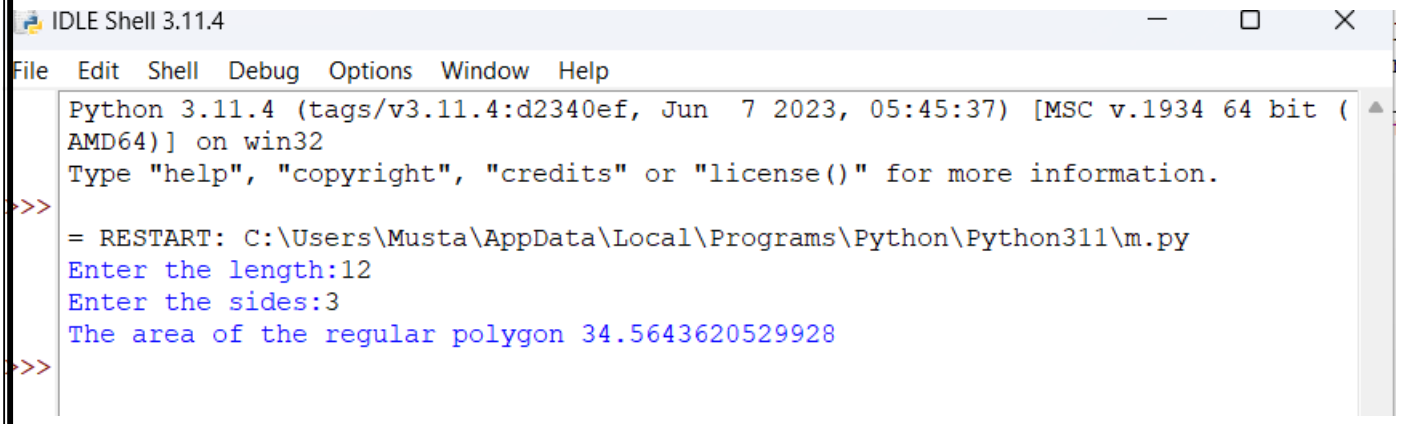
ALGORITHM:

Step1:Start
Step1:import the math module
Step1:Get input from the user and store in length
Step1:Get the input from the user for number of sides
Step1:And now store length and number of sides in l and n
Step6:And now the apply the area of polygon formula and import it with math Module
Step1:and store it in area of polygon
Step1:Stop

PROGRAM:

```
#[l2n]/[4tan(180/n)]
import math
length = int(input('Enter the length:'))
num_of_sides = int(input("Enter the sides:"))
l = length
n = num_of_sides
area_polygon = (l**2)*n/4*math.tan(180/n)
print("The area of the regular polygon",area_polygon)
```

OUTPUT:



Ex. No:	86	Python program to calculate the square of each element in a list
Date:		

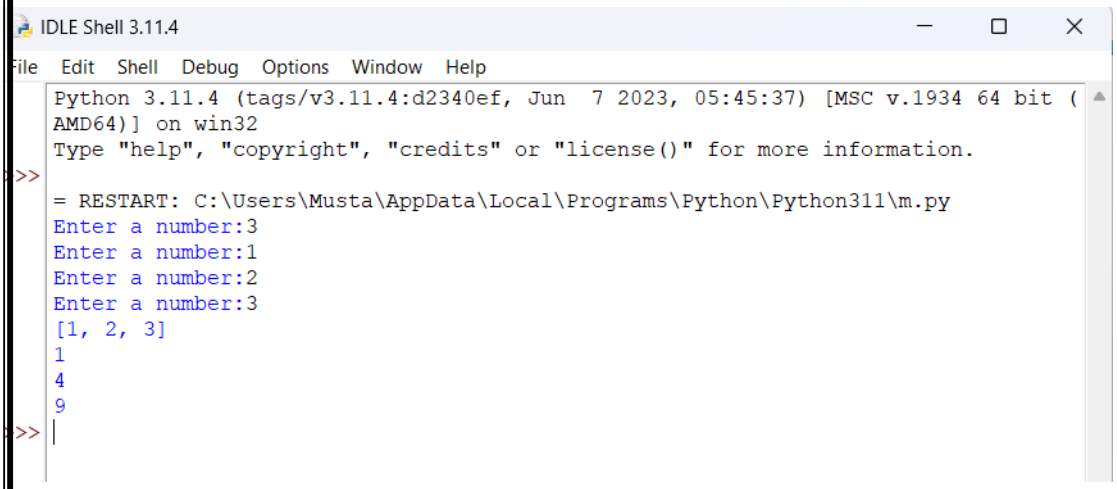
ALGORITHM:

Step1:Start
Step1:Get the value of the list from the user and store in the lst
Step1:print lst
Step1:Take the other variable x and take the value as 1
Step1:Compute the loop of the list and store the value in i
Step1:and take the condition as I double modulus of 2 and store the value in x
Step1:print x
Step1:Stop

PROGRAM:

```
lst = []
number_of_elements  = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
x = 1
for i  in lst:
  x= i**2
  print(x)
```

OUTPUT:



Ex. No:	87	program to create and access elements in a tuple
Date:		

ALGORITHM:

Step1:Start
Step1:Get the value and store in tup_1
Step1:print tup_1
Step1:Stop

PROGRAM:

```
tup_1 = (1,2,3,4,5)
print(tup_1)
```

OUTPUT:

```
>>>
===== RESTART: C:\Users\Musta\AppData\Local\Programs\Python\Python311\m.py =====
(1, 2, 3, 4, 5)
>>>
```

Ex. No:	88	program to create and access elements in a list
Date:		

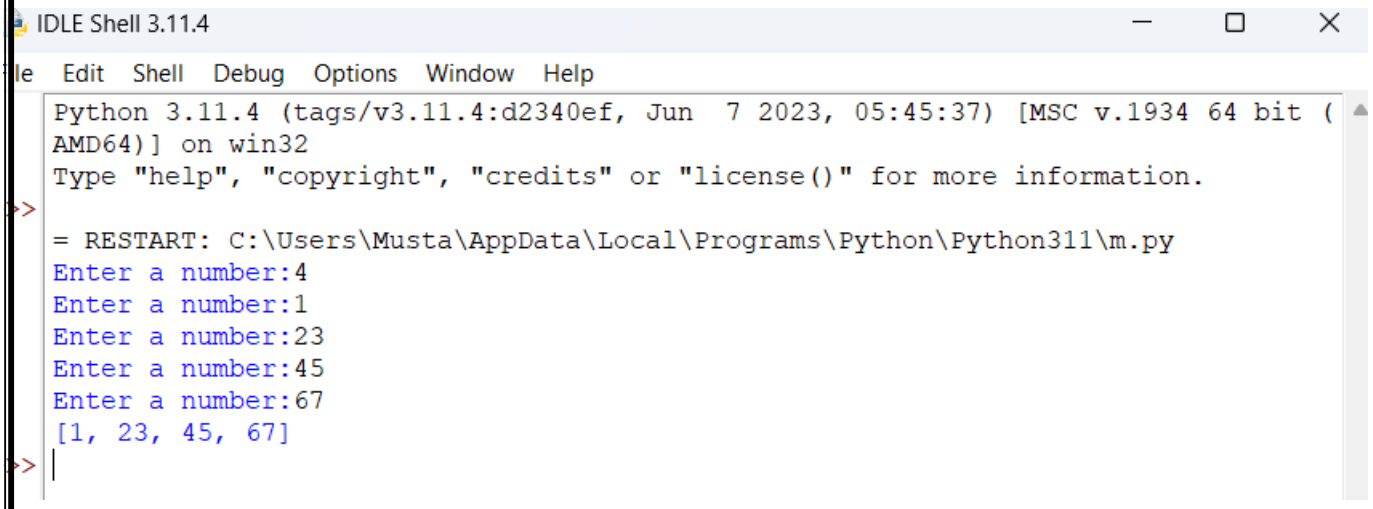
ALGORITHM:

Step1:Start
Step1:Take the empty list and store in lst
Step1:Take the input from user for the value of iteration and store in number_of_elements
Step1:Compute the loop of number_of_elements and store in i
Step1:And inside the loop take the input from the user and store in element
Step1:and append the value of element one by one in a variable lst with method Of append
Step1:Stop

PROGRAM:

```
lst = []
number_of_elements  = int(input("Enter a number:"))
for i in range(number_of_elements):
  element = int(input("Enter a number:"))
  lst.append(element)
print(lst)
```

OUTPUT:



Ex. No:	89	Python program to check if a given string is alphanumeric.
Date:		

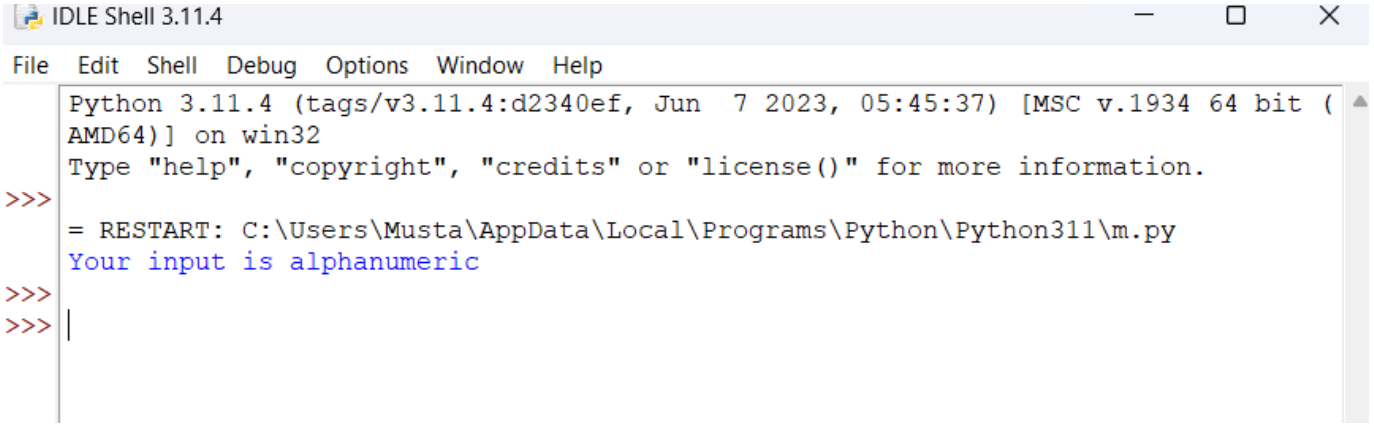
ALGORITHM:

Step1:Start
Step1:Get the input from the user
Step1:check the condition input is equal to the input function isalnum()
Step1:print your number is not alphanumeric
Step1:else print your number is alphanumeric
Step1:Stop

PROGRAM:

```
num = eval(input("Enter a number"))
if num == num.isalnum():
  print("Your input is not alphanumeric")
else:
  print("Your input is alphanumeric")
```

OUTPUT:



Ex. No:	90	Python program to calculate the average of a list of floats
Date:		

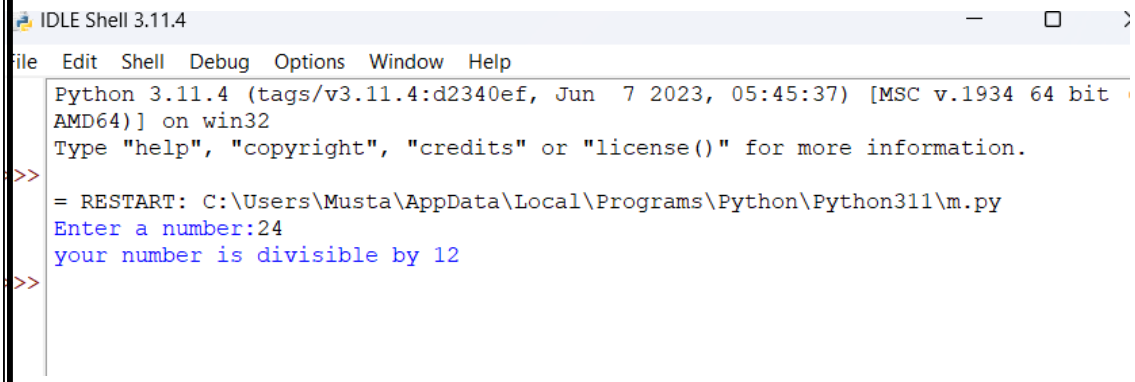
ALGORITHM:

Step1:Start
Step1:Get the list of the values of floats
Step1:Compute the loop in list and store in the variable i
Step1:and take the other variable x
Step1:Compute the condition i divided by 2 and store in x
Step1:print x
Step1:Stop

PROGRAM:

```
lst = [11.3,24.6,467.5,87.7]
for i in lst:
  x = i//2
  print(x)
```

OUTPUT:



Ex. No:	91	Python program to find the cube of a number
Date:		

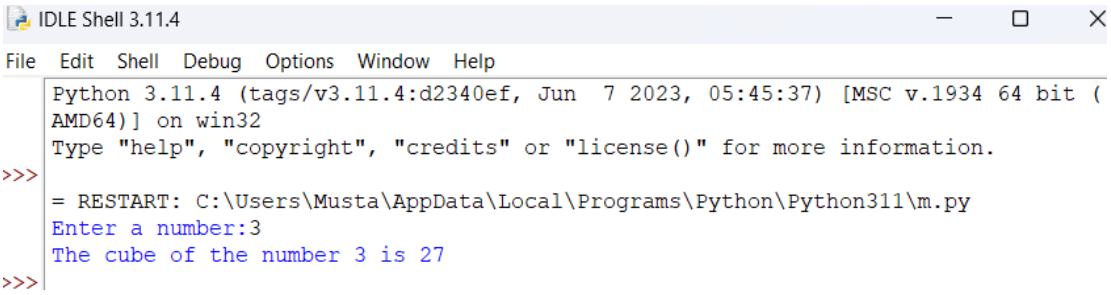
ALGORITHM:

Step1:Start
Step1:Get the input from the user
Step1:print the cube of the number is input3**
Step1:Stop

PROGRAM:

```
num = int(input("Enter a number:"))
print("The cube of the number",num,"is",num**3)
```

OUTPUT:



Ex. No:	92	Python program to calculate the perimeter of a rectangle.
Date:		

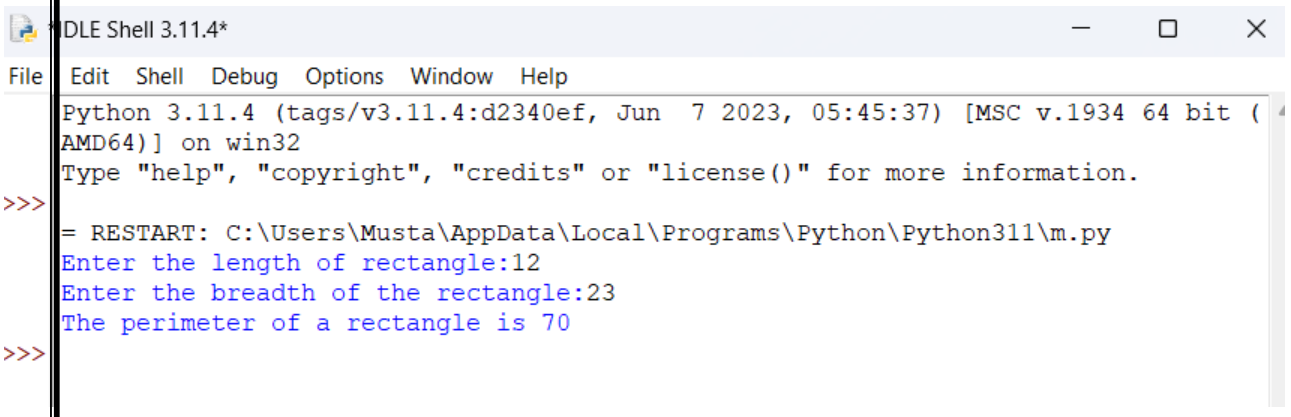
ALGORIHTM:

Step1:Get the input for the length of the rectangle
Step1:Get the input for the breadth of the rectangle
Step1:and compute perimeter is equal to 2*(length + breadth)
Step1:print perimeter
Step1:Stop

PROGRAM:

```
length = int(input("Enter the length of rectangle:"))
breadth = int(input("Enter the breadth of the rectangle:"))
perimeter = 2*(length + breadth)
print("The perimeter of a rectangle is",perimeter)
```

OUTPUT:



Ex. No:	93	python program to print factorial
Date:		

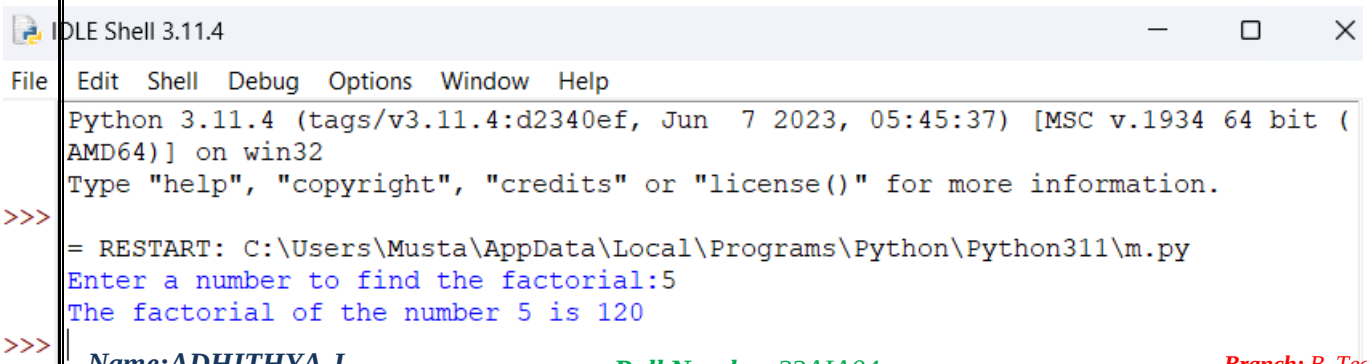
ALGORITHM:

Step1:Start
Step1:Get the input from the user
Step1: Take the other variable x and the store the value 1
Step1:Compute the loop from 1 to the input and the store the value in i
Step1:and keep the other condition as x*i
Step1:print the value of x
Step1:Stop

PROGRAM:

```
num = int(input("Enter a number to find thefactorial:"))
x= 1
for i in range(1, num+1):
  x = x*i
print("The factorial of the
number",num,"is",x)
```

OUTPUT:



Ex. No:	94	Python program to print the elements of a list one by one.
Date:		

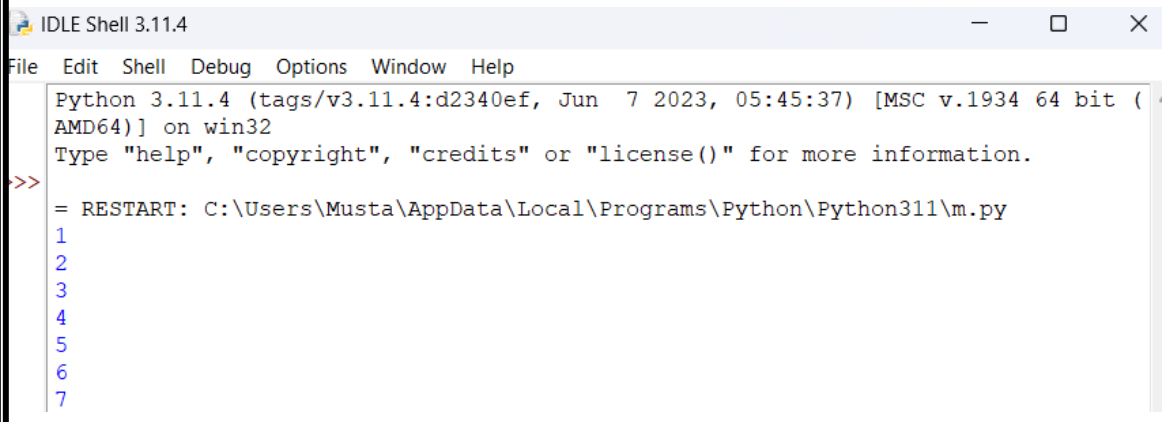
ALGORITHM:

Step1:Start
Step1: Take the list as [1,2,3,4,5,6,7]
Step1:Run the loop with list and store it in i
Step1:print i
Step1:Stop

PROGRAM:

```
lst = [1,2,3,4,5,6,7]
for i in lst:
  print(i)
```

OUTPUT:



Ex. No:	95	Python program to print the square numbers between 1 and 10
Date:		

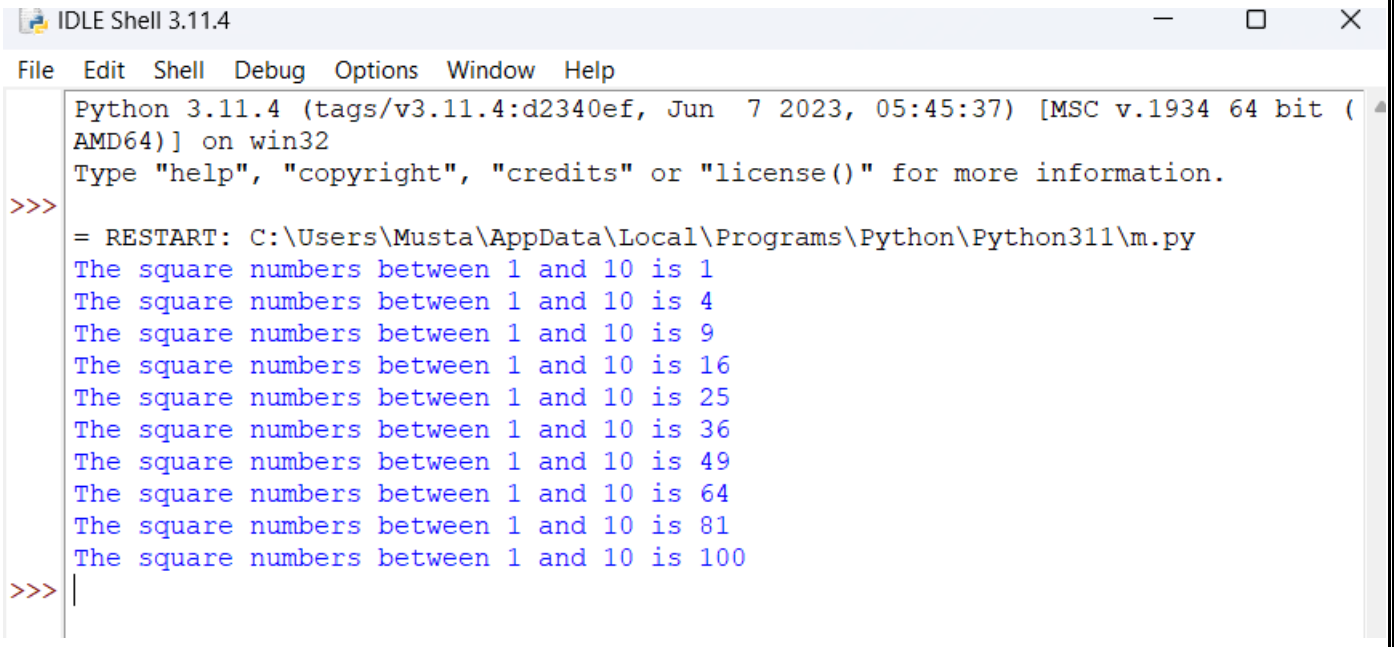
ALGORITHM:

Step1:Start
Step1: get the first number as 1
Step1:get the last number as 10
Step1:compute the loop from first to last and store in i
Step1:and keep the condition as x is equal to i2**
Step1:Stop

PROGRAM:

```
first = 1
last = 10
for i in range(first,last+1):
  x = i**2
  print("The square numbers between 1 and 10 is",x)
```

OUTPUT:



Ex. No:	96	Python program to check if a given character is a digit
Date:		

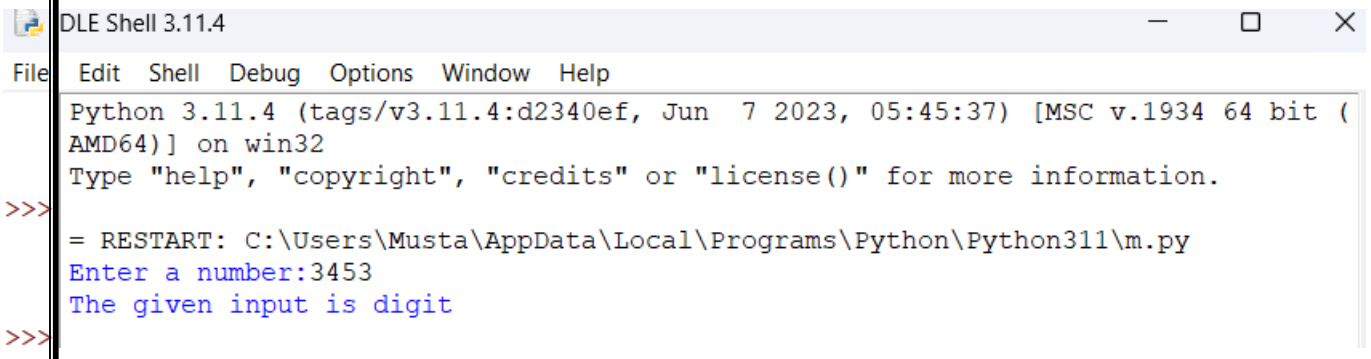
ALGORITHM:

Step1:Start
Step1:Get the input from the user
Step1:Check if the input is equal to the num.isdigit()
Step1:print the given input Is not a digit
Step1:else print the given input is not a digit
Step1:stop

PROGRAM:

```
num = input("Enter a number:")
if num == num.isdigit():
   print("The given input is not digit")
else:
   print("The given input is digit")
```

OUTPUT:



Ex. No:	97	Python program to calculate the average of three numbers entered by the user.
Date:		

ALGORITHM:

Step1:start
Step1:Get the input from the user and store in num1
Step1:Get the input from the user and store in num2
Step1:Get the input from the user and store in num3
Step5:Compute all the three inputs and divided by 2
Step1:print all the inputs
Step1:Stop

PROGRAM:

```
num_1 = int(input("Enter a number:"))
num_2 = int(input("Enter a number:"))
num_3 = int(input("Enter a number:"))
print("num_1=",num_1//2)
print("num_2=",num_2//2)
print("num_3=",num_3//2)
```

OUTPUT:

```
>>
===== RESTART: C:\Users\Musta\AppData\Local\Programs\Python\Python311\m.py =====
Enter a number:434
Enter a number:534
Enter a number:754
num_1= 217
num_2= 267
num_3= 377
>>
```

Ex. No:	98	Python program to check if a number is a multiple of 2
Date:		

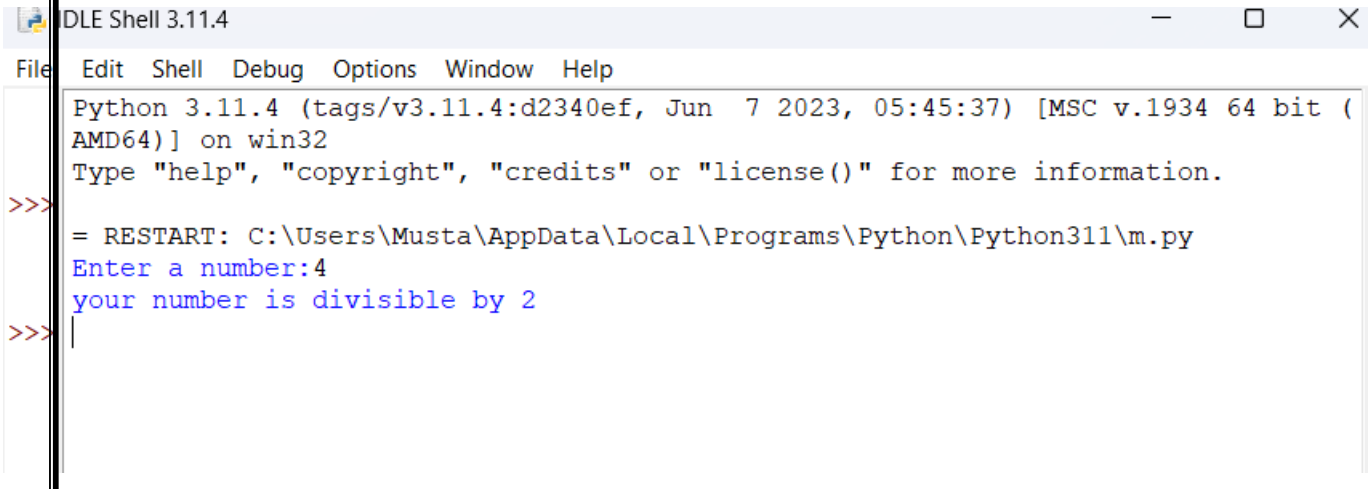
ALGORITHM:

Step1:start
Step1:Get the input from the user
Step1:check the condition if input is divisibble by 2
Step1:print input is divisible by 2
Step1:otherwise print input is not divisiblle bby 2
Step1:stop

PROGRAM:

```
num = int(input("Enter a number:"))
if num%2 == 0:
  print("your number is divisible by 2")
else:
  print("your number is not divisible by 2")
```

OUTPUT:



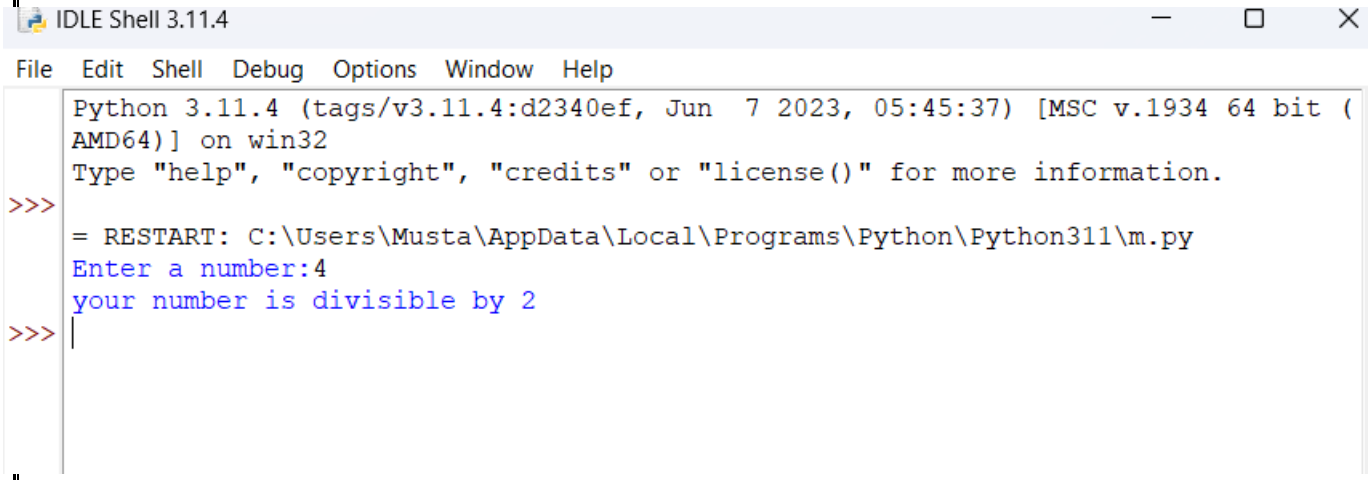
Ex. No:	99	Python program to check if a number is a multiple of 5
Date:		

ALGORITHM:

Step1:start
Step1:Get the input from the user
Step1:check the condition if input is divisibble by 5
Step1:print input is divisible by 5
Step1:otherwise print input is not divisiblle bby 5
Step1:stop

PROGRAM:

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

num%5 == 0:

  print("your number is divisible by 5")else:

  print("your number is not divisible by 5OUTPUT:
```



Ex. No:	100	Python program to check if a number is a multiple of 12
Date:		

ALGORITHM:

Step1:start
Step1:Get the input from the user
Step1:check the condition if input is divisibble by 12
Step1:print input is divisible by 12
Step1:otherwise print input is not divisiblle bby 12
Step1:stop

PROGRAM:

```
num = int(input("Enter a number:"))
ibl
  print("your number is divisible by 12")
else:
  print("your number is not divisible by 12")
```

OUTPUT: