

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-1

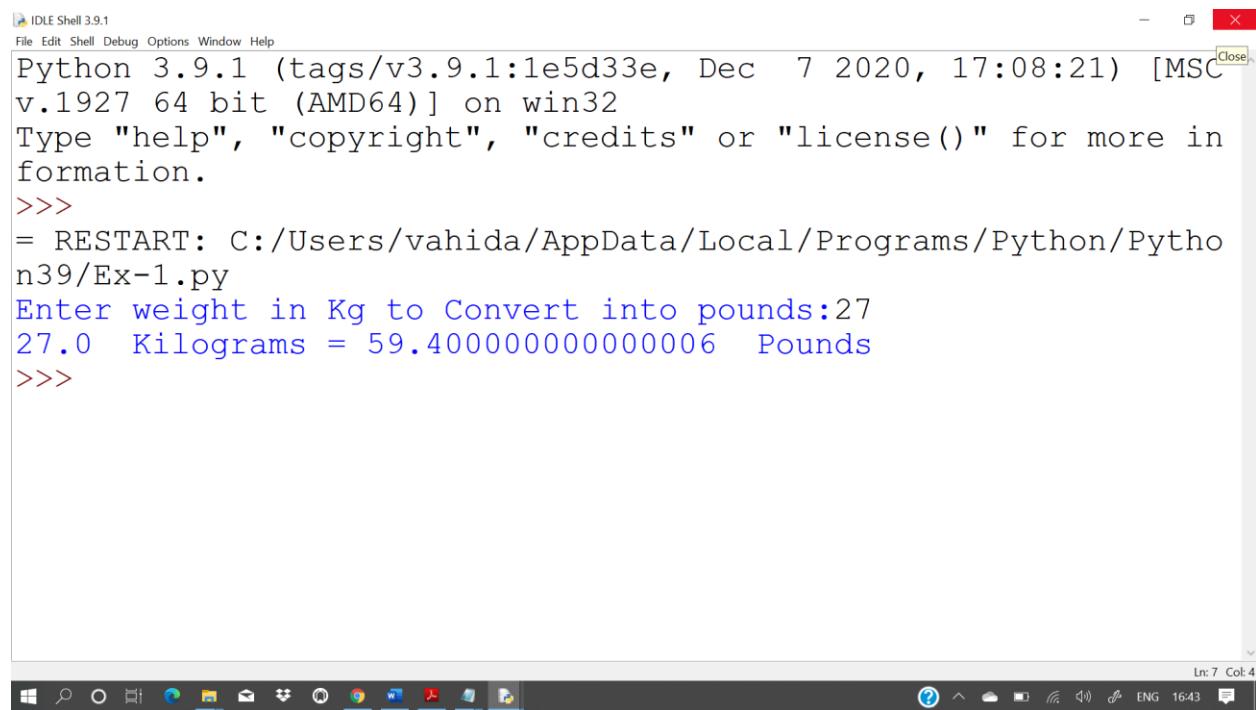
AIM:

Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.

PROGRAM:

```
weight_in_kilo_grams = float(input('Enter weight in Kg to Convert into pounds:'))  
pounds = weight_in_kilo_grams * 2.2  
print(weight_in_kilo_grams,' Kilograms =', pounds,' Pounds')
```

OUTPUT:



The screenshot shows a Python IDLE Shell window. The title bar reads "IDLE Shell 3.9.1". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main window displays the following text:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/Ex-1.py  
Enter weight in Kg to Convert into pounds:27  
27.0 Kilograms = 59.40000000000006 Pounds  
>>>
```

The status bar at the bottom right shows "Ln: 7 Col: 4". The taskbar at the bottom of the screen shows various application icons.

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-2

AIM:

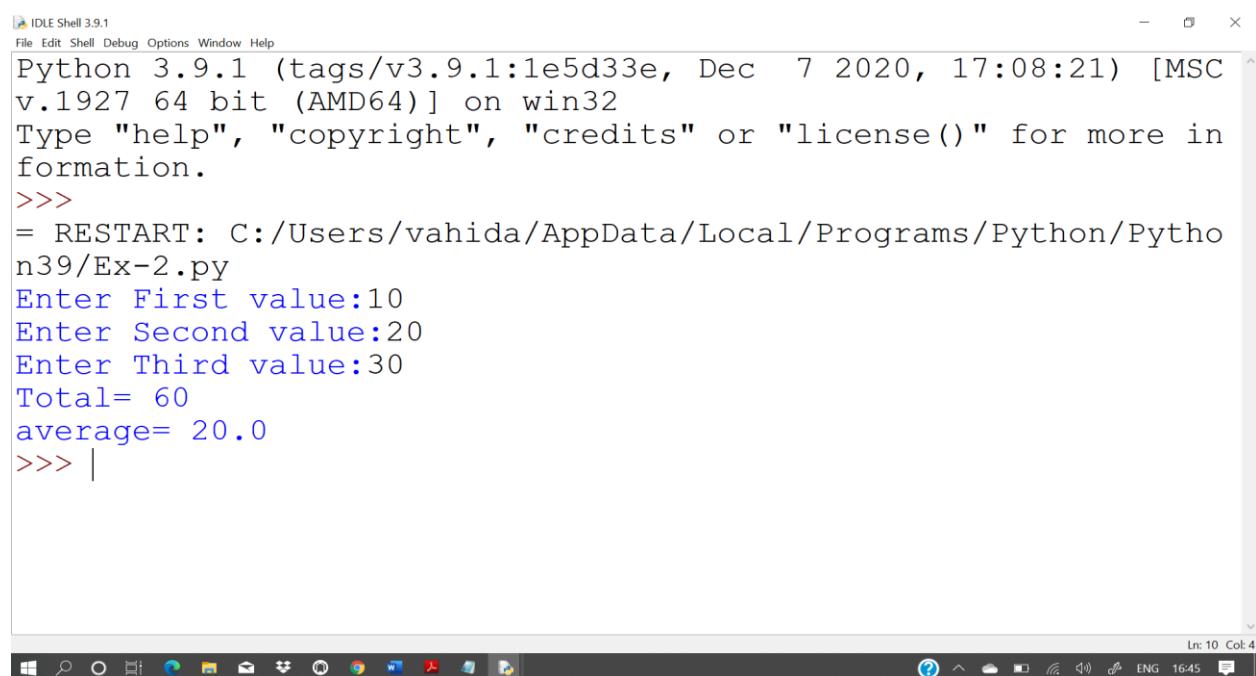
Write a program that asks the user to enter three numbers (use three separate input statements).

Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average.

PROGRAM:

```
a=int(input('Enter First value:'))  
b=int(input('Enter Second value:'))  
c=int(input('Enter Third value:'))  
  
sum=a+b+c  
  
average=(a+b+c)/3  
  
print("Total=",sum)  
  
print("average=",average)
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/Ex-2.py  
Enter First value:10  
Enter Second value:20  
Enter Third value:30  
Total= 60  
average= 20.0  
>>> |
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-3

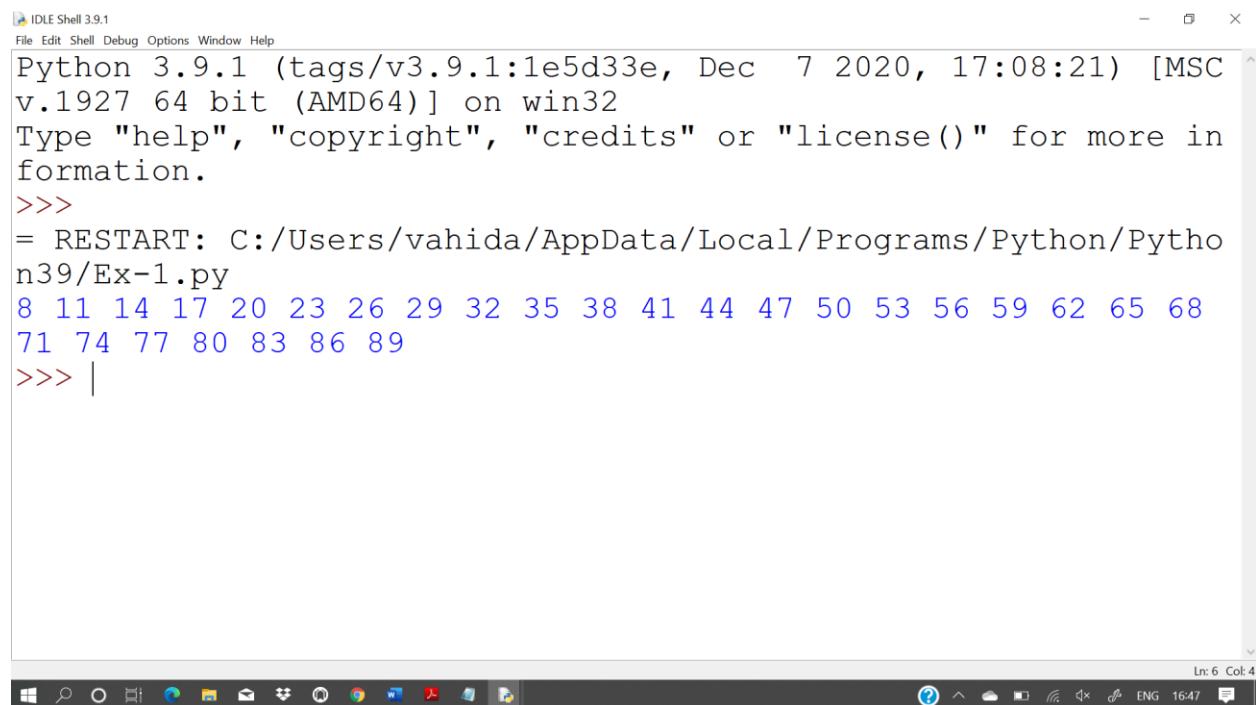
AIM:

Write a program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.

PROGRAM:

```
for i in range(8,90,3):
    print(i,end=" ")
```

OUTPUT:



The screenshot shows a Windows desktop environment with a Python IDLE Shell window open. The window title is "IDLE Shell 3.9.1". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell area displays the following text:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/Ex-1.py
8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68
71 74 77 80 83 86 89
>>> |
```

The taskbar at the bottom shows various icons for system functions like search, file explorer, and network. The status bar at the bottom right indicates "Ln: 6 Col: 4".

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-4

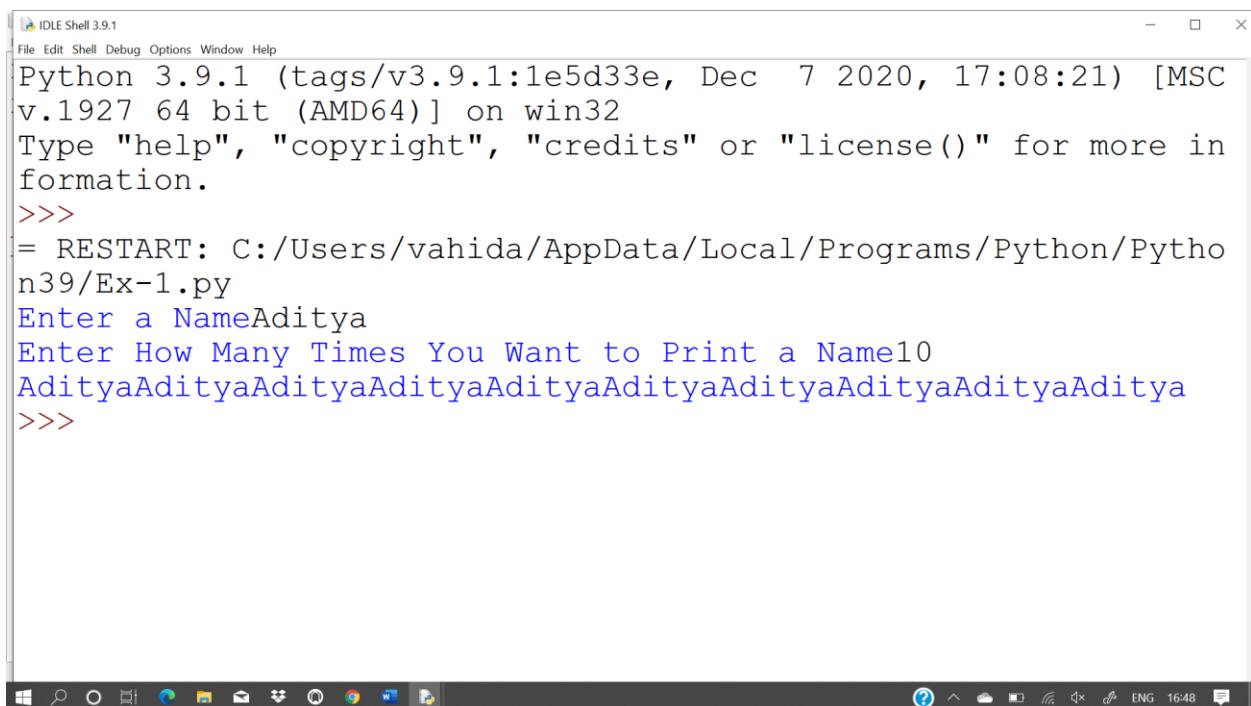
AIM:

Write a program that asks the user for their name and how many times to print it. The program should print out the user name the specified number of times.

PROGRAM:

```
name=input("Enter a Name")  
n=int(input("Enter How Many Times You Want to Print a Name"))  
print(name*n)
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/Ex-1.py  
Enter a NameAditya  
Enter How Many Times You Want to Print a Name10  
AdityaAdityaAdityaAdityaAdityaAdityaAdityaAdityaAdityaAditya  
>>>
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-5

AIM:

Use a for loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.

*

**

PROGRAM:

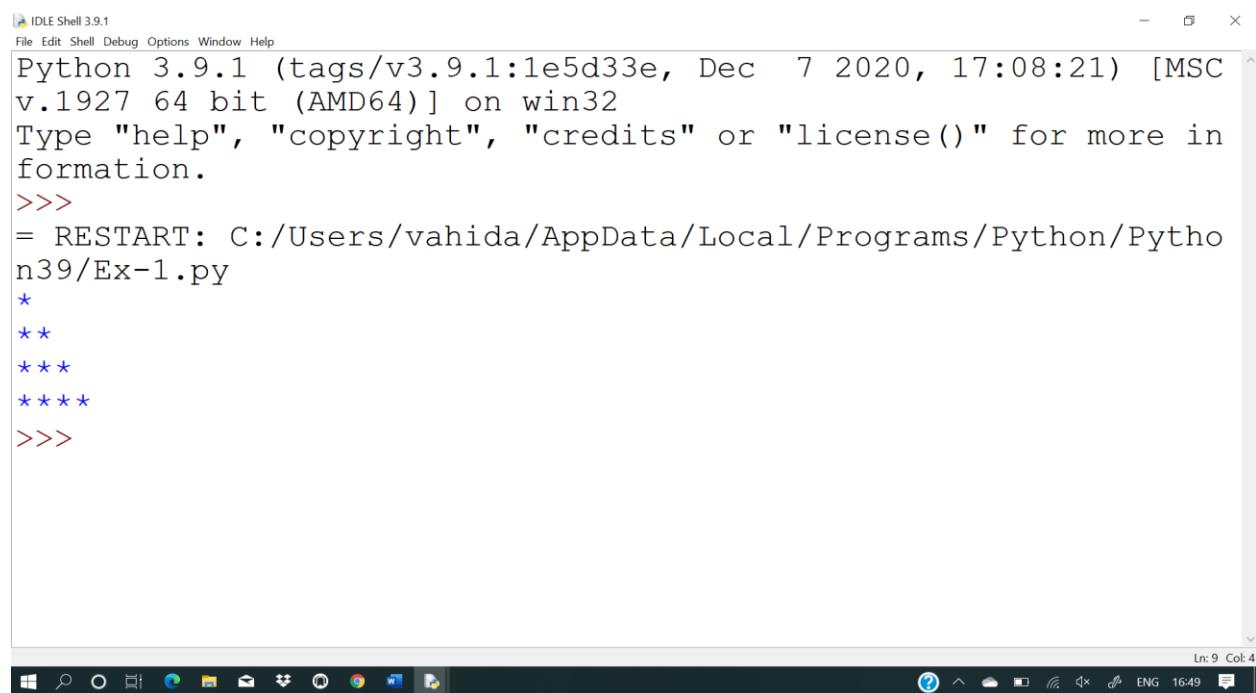
```
for i in range(1,5):
```

```
    for j in range(i):
```

```
        print('*',end="")
```

```
    print()
```

OUTPUT:



The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.9.1". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following text:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/Ex-1.py
*
**
***
>>>
```

The status bar at the bottom right shows "Ln: 9 Col: 4". The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, Edge, Mail, and File Explorer.

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-6

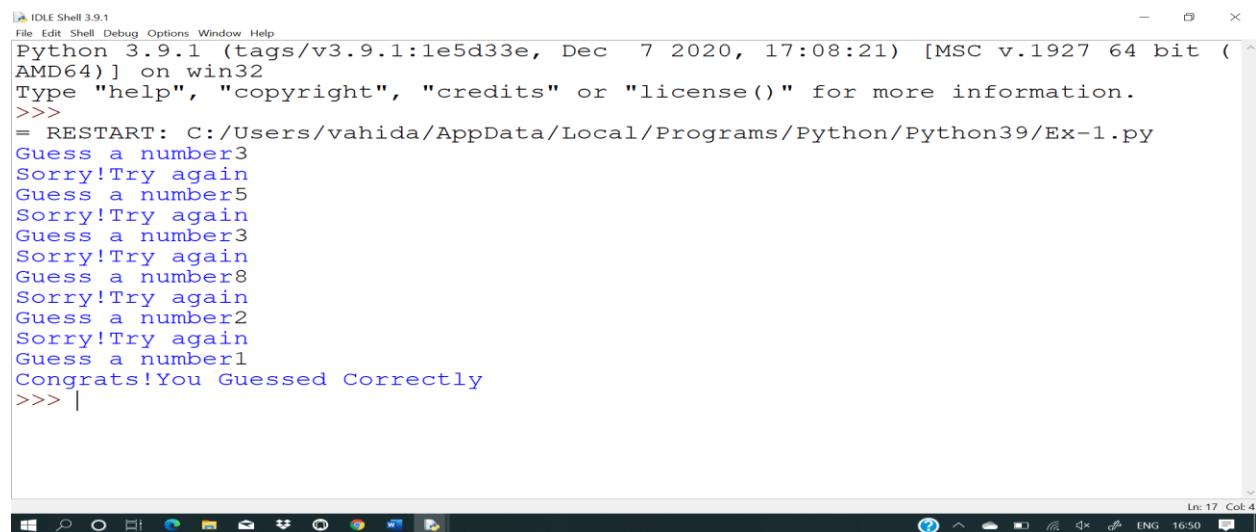
AIM:

Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.

PROGRAM:

```
import random  
  
r=random.randint(1,10)  
  
while True:  
  
    guess=int(input("Guess a number"))  
  
    if r==guess:  
  
        print("Congrats!You Guessed Correctly")  
  
        break  
  
    else:  
  
        print("Sorry!Try again")
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/Ex-1.py  
Guess a number3  
Sorry! Try again  
Guess a number5  
Sorry! Try again  
Guess a number3  
Sorry! Try again  
Guess a number8  
Sorry! Try again  
Guess a number2  
Sorry! Try again  
Guess a number1  
Congrats! You Guessed Correctly  
>>> |
```

EXERCISE-7

AIM:

Write a program that asks the user for two numbers and prints Close if the numbers are within .001 of each other and Not close otherwise.

PROGRAM:

```
n1=float(input("Enter First Number: "))

n2=float(input("Enter Second Nuuber: "))

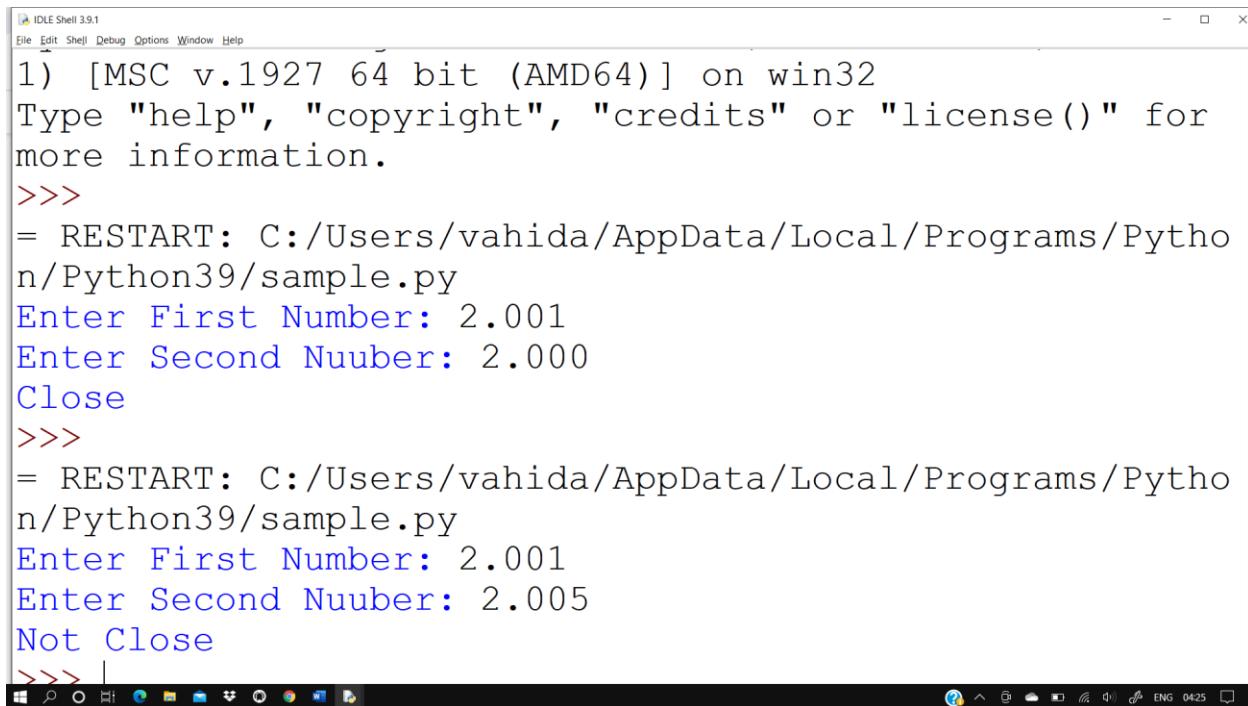
if abs(n1-n2)<=0.001:

    print("Close")

else:

    print("Not Close")
```

OUTPUT:



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help

1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.

>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter First Number: 2.001
Enter Second Nuuber: 2.000
Close
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter First Number: 2.001
Enter Second Nuuber: 2.005
Not Close
>>>
```

EXERCISE-8

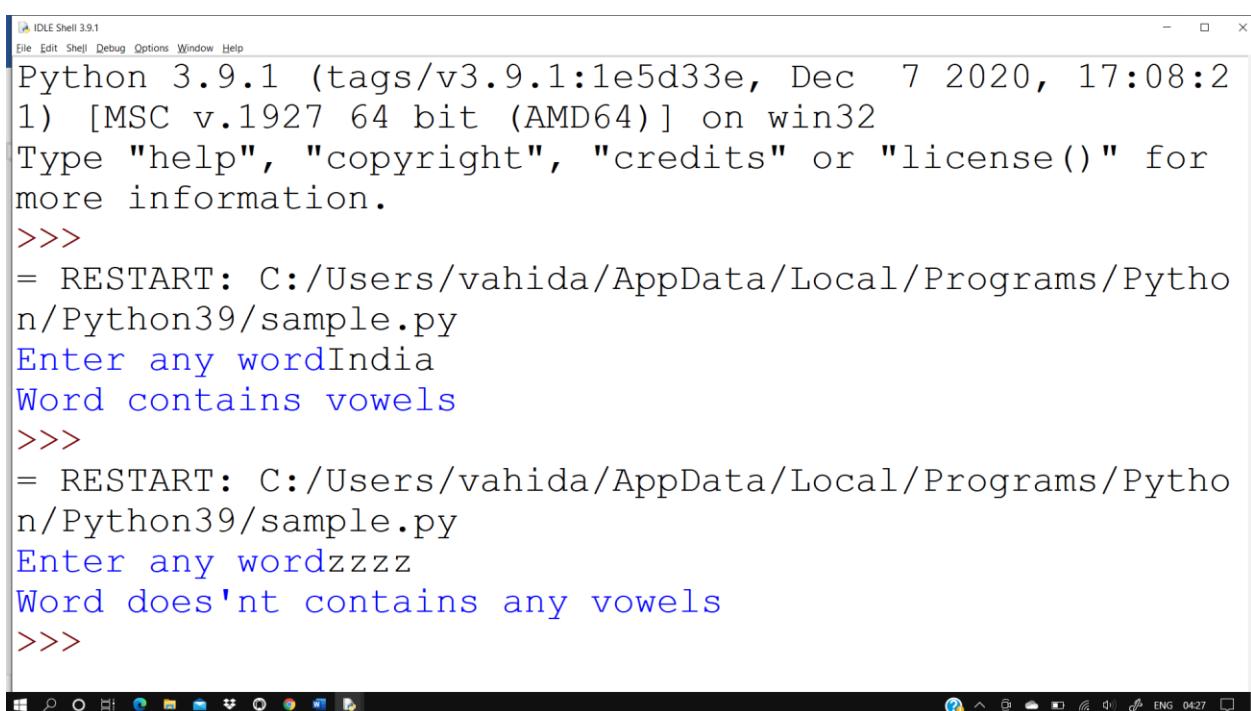
AIM:

Write a program that asks the user to enter a word and prints out whether that word contains any vowels.

PROGRAM:

```
word=input("Enter any word")
status=False
for i in word:
    if i in "aeiouAEIOU":
        status=True
if status==True:
    print("Word contains vowels")
else:
    print("Word does'nt contains any vowels")
```

OUTPUT:



The screenshot shows the IDLE Shell 3.9.1 interface. The window title is "IDLE Shell 3.9.1". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main area displays the Python interpreter's welcome message and two sessions of code execution. In the first session, the user enters "India" and the output is "Word contains vowels". In the second session, the user enters "zzzz" and the output is "Word does'nt contains any vowels". The system tray at the bottom shows standard icons for network, battery, and system status.

```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.

>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho
n/Python39/sample.py
Enter any wordIndia
Word contains vowels
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho
n/Python39/sample.py
Enter any wordzzzz
Word does'nt contains any vowels
>>>
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-9

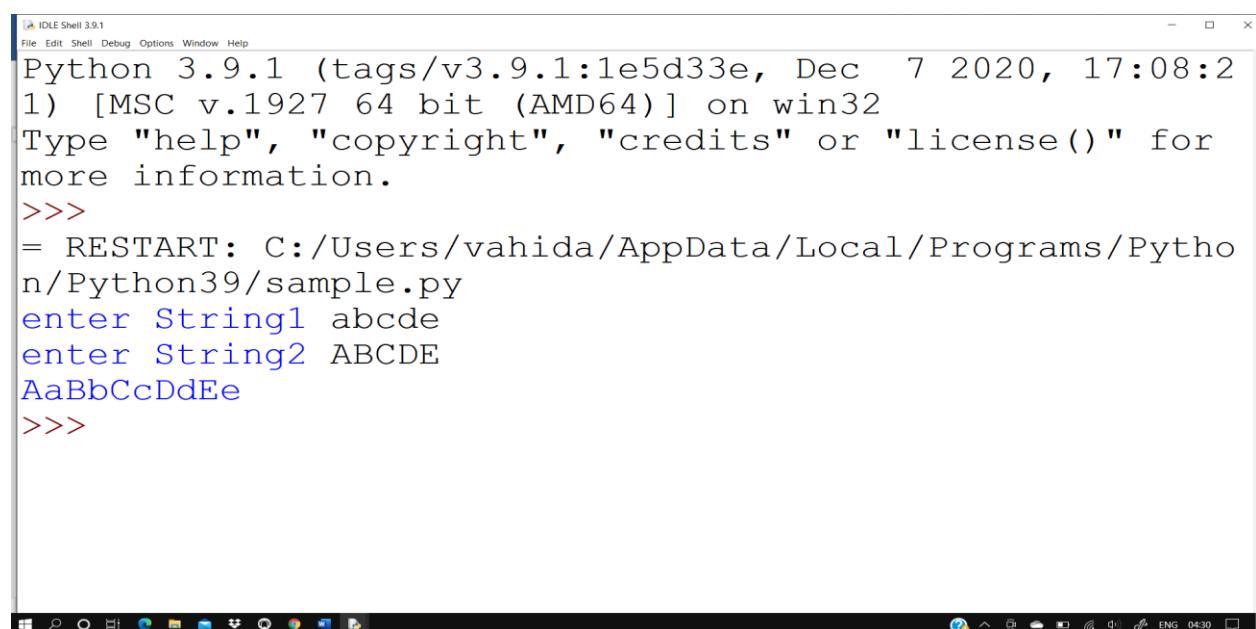
AIM:

Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe.

PROGRAM:

```
s1=input("enter String1 ")
s2=input("enter String2 ")
if len(s1)!=len(s2):
    print("Length Of Two Strings Are Not Equal!Try Later")
    exit(0)
else:
    r=""
    for i in range(len(s1)):
        r=r+s2[i]+s1[i]
    print(r)
```

OUTPUT:



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
enter String1 abcde
enter String2 ABCDE
AaBbCcDdEe
>>>
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-10

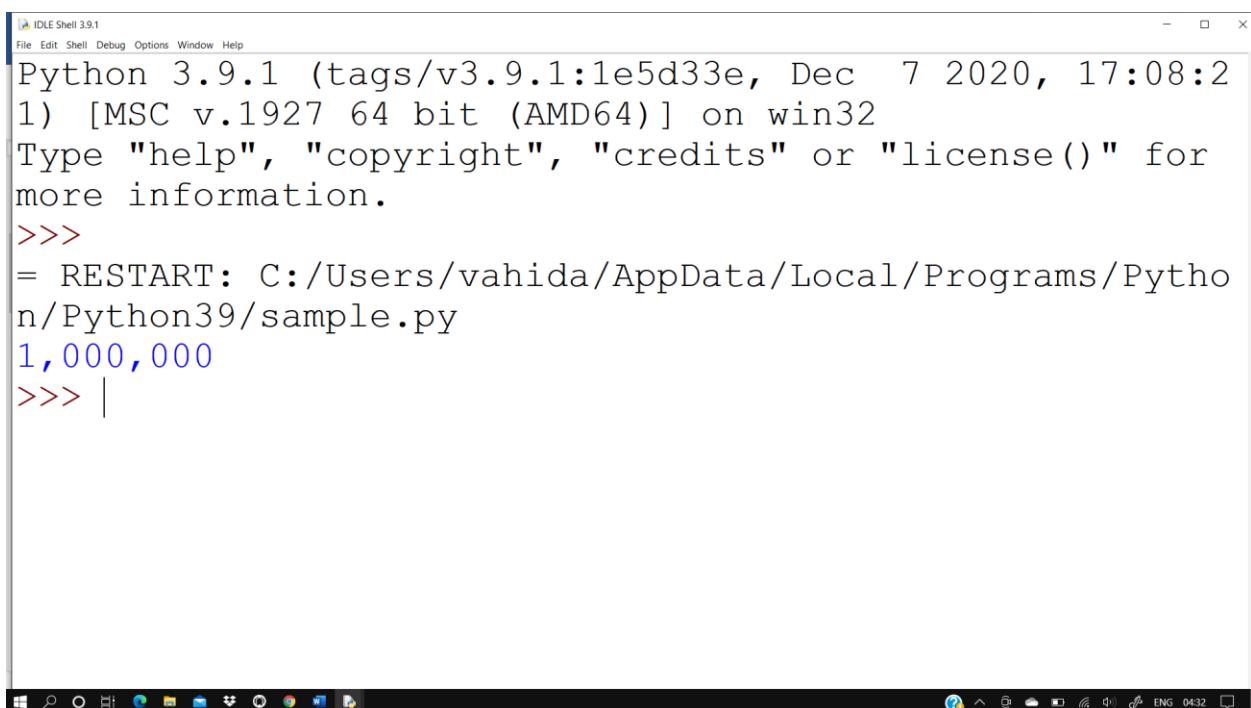
AIM:

Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be 1,000,000.

PROGRAM:

```
number_with_commas = "{:,}").format(1000000)  
print(number_with_commas)
```

OUTPUT:



The screenshot shows a Windows desktop environment with a Python IDLE Shell window open. The window title is "IDLE Shell 3.9.1". The shell displays the following text:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2  
1) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for  
more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py  
1,000,000  
>>> |
```

The taskbar at the bottom shows various icons for system applications like File Explorer, Task View, and Control Panel. The system tray indicates the date and time as "ENG 04:32".

EXERCISE-11

AIM:

In algebraic expressions, the symbol for multiplication is often left out, as in $3x+4y$. Computers prefer those expressions to include the multiplication symbol, like $3*x+4*y$. Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate.

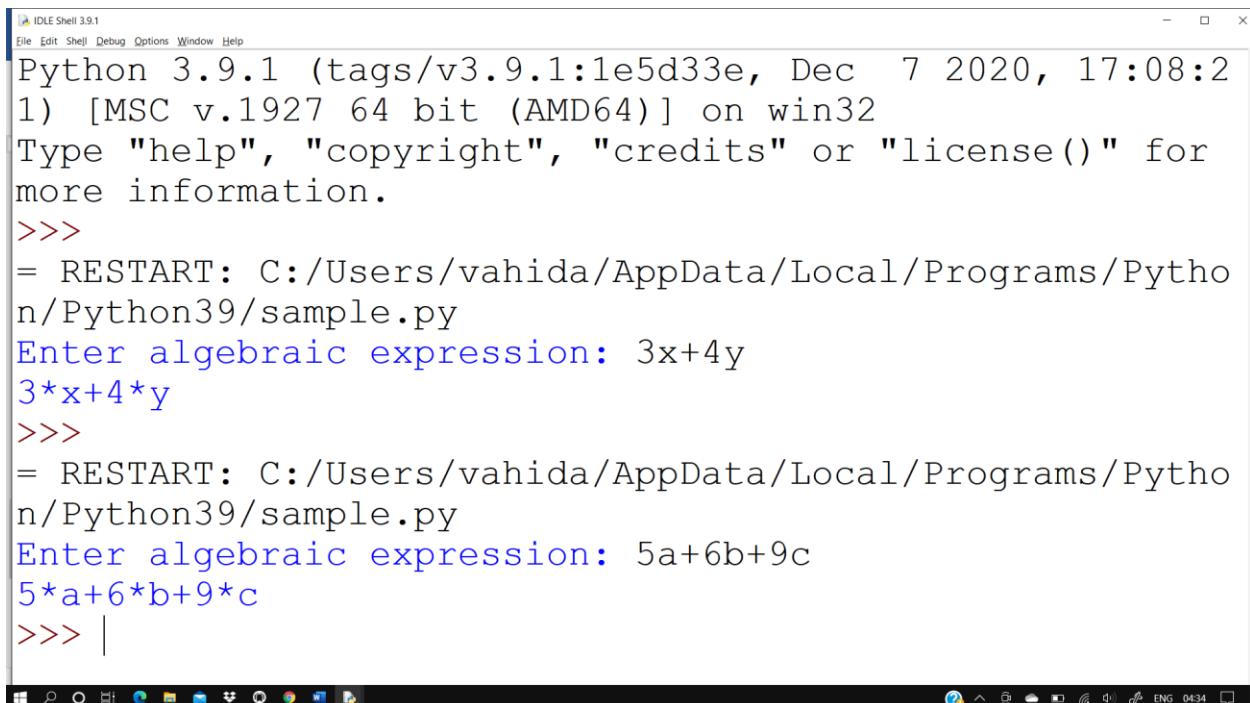
PROGRAM:

```
s=input("Enter algebraic expression: ")  
l=list(s)  
result=""  
i=0  
while(i<len(l)):  
    if l[i]=='(':  
        index=l.index(')')  
        s2=".join(l[i:index+1])  
        result=result+'*'+s2  
        i=i+len(s2)  
    elif l[i].isalpha():  
        result=result+'*'+l[i]  
        i=i+1  
    else:  
        result=result+l[i]  
        i=i+1
```

PYTHON PROGRAMMING LAB MANUAL

```
print(result)
```

OUTPUT:



The screenshot shows two separate sessions in the Python IDLE Shell 3.9.1. The first session starts with the Python version information and a prompt for an algebraic expression. The user enters `3*x+4*y`, which is highlighted in blue. The second session starts with a restart, followed by another prompt for an algebraic expression. The user enters `5*a+6*b+9*c`, which is also highlighted in blue. Both sessions end with a final prompt.

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21)
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.

>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter algebraic expression: 3x+4y
3*x+4*y
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter algebraic expression: 5a+6b+9c
5*a+6*b+9*c
>>> |
```

EXERCISE-12

AIM:

Write a program that generates a list of 20 random numbers between 1 and 100.

- (a) Print the list.
- (b) Print the average of the elements in the list.
- (c) Print the largest and smallest values in the list.
- (d) Print the second largest and second smallest entries in the list
- (e) Print how many even numbers are in the list.

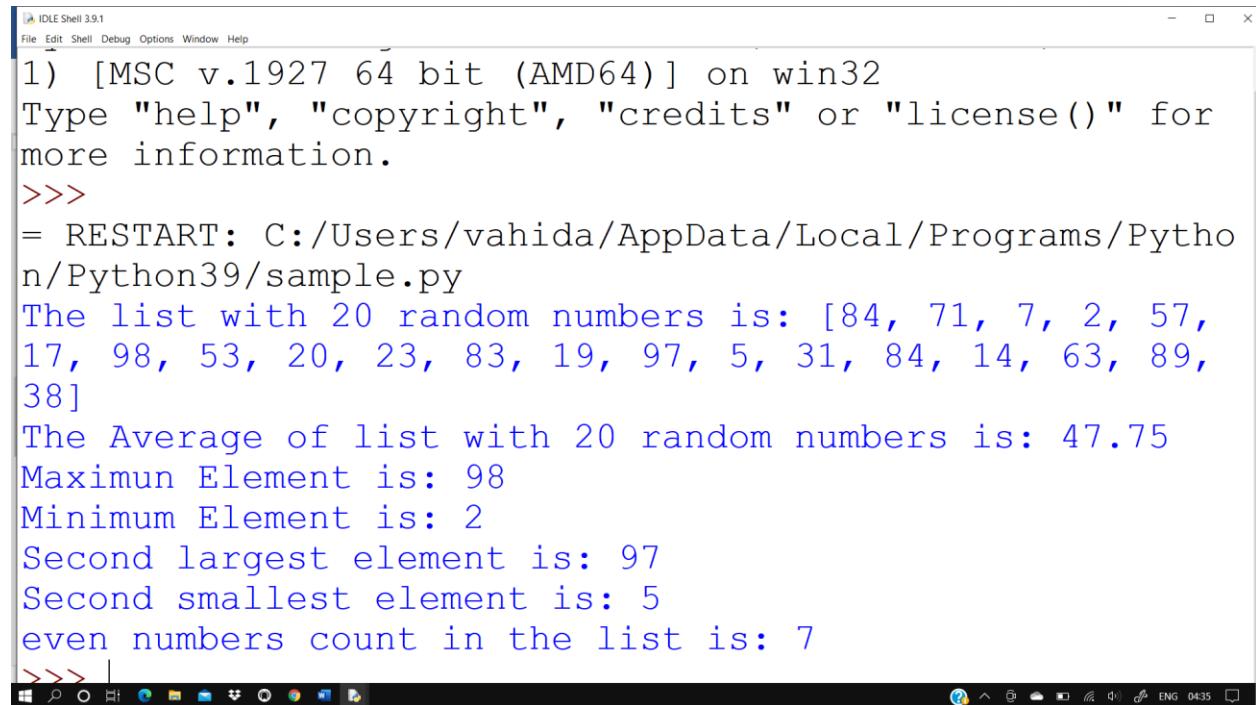
PROGRAM:

```
import random  
r=[]  
for i in range(20):  
    r.append(random.randint(1,100))  
  
# (a) Print the list.  
  
print("The list with 20 random numbers is:",r)  
  
# (b) Print the average of the elements in the list.  
  
print("The Average of list with 20 random numbers is:",round(sum(r)/len(r),2))  
  
# (c) Print the largest and smallest values in the list.  
  
print("Maximun Element is:",max(r))  
  
print("Minimum Element is:",min(r))  
  
#(d) Print the second largest and second smallest entries in the list  
  
r1=list(set(r))  
r1.sort()
```

PYTHON PROGRAMMING LAB MANUAL

```
print("Second largest element is:",r1[-2])  
print("Second smallest element is:",r1[1])  
#(e) Print how many even numbers are in the list.  
  
even=[i for i in r if i%2==0]  
  
print("even numbers count in the list is:",len(even))
```

OUTPUT:



The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.9.1". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main window displays the following Python session:

```
1) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for  
more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py  
The list with 20 random numbers is: [84, 71, 7, 2, 57,  
17, 98, 53, 20, 23, 83, 19, 97, 5, 31, 84, 14, 63, 89,  
38]  
The Average of list with 20 random numbers is: 47.75  
Maximun Element is: 98  
Minimum Element is: 2  
Second largest element is: 97  
Second smallest element is: 5  
even numbers count in the list is: 7  
>>>
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-13

AIM:

Write a program that asks the user for an integer and creates a list that consists of the factors of that integer.

PROGRAM:

```
num=int(input("Enter an integer: "))

factors=[]

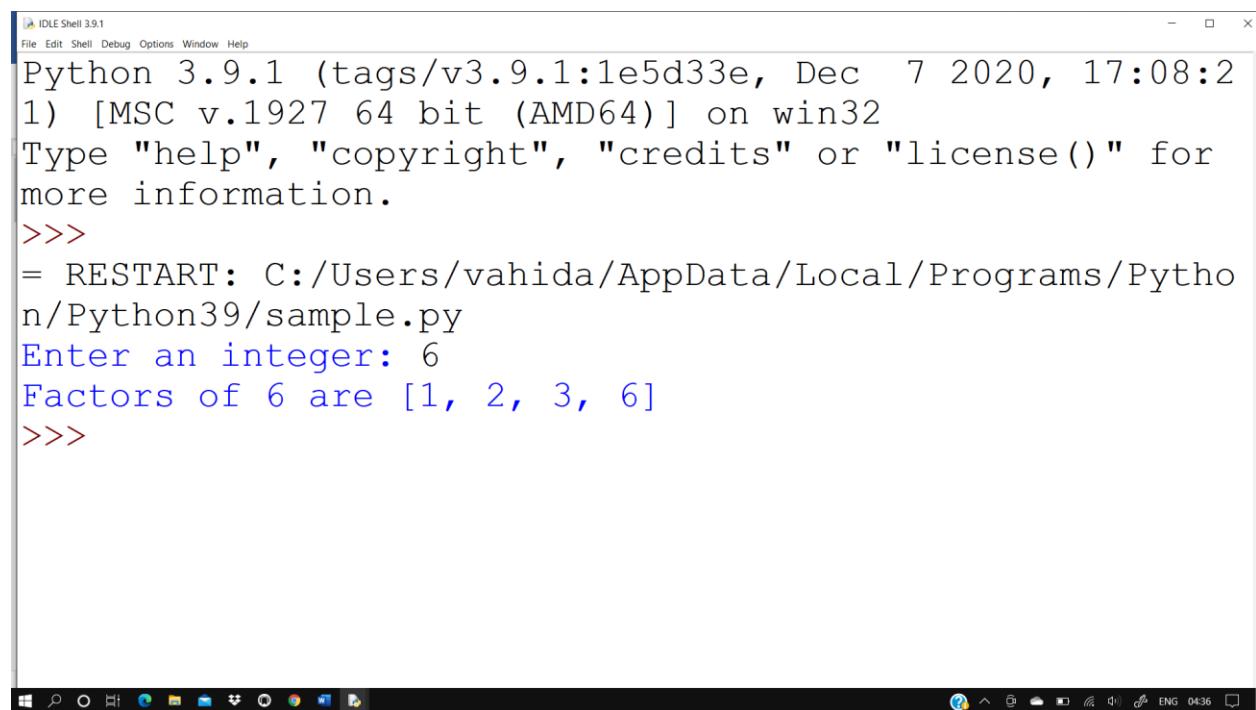
for i in range(1,num+1):

    if num%i==0:

        factors.append(i)

print("Factors of",num,"are",factors)
```

OUTPUT:



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.

>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho
n/Python39/sample.py
Enter an integer: 6
Factors of 6 are [1, 2, 3, 6]
>>>
```

EXERCISE-14

AIM:

Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0]

PROGRAM:

```
l=[int(i) for i in input().split()]
```

```
result=[]
```

```
for i in l:
```

```
    if i not in result:
```

```
        result.append(i)
```

```
print(result)
```

OUTPUT:

The screenshot shows the Python IDLE Shell 3.9.1 interface. The window title is "IDLE Shell 3.9.1". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main console area displays the following output:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:2  
1) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for  
more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho  
n/Python39/sample.py  
1 1 2 3 4 3 0 0  
[1, 2, 3, 4, 0]  
>>> |
```

The taskbar at the bottom of the screen shows various application icons.

EXERCISE-15**AIM:**

Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.

PROGRAM:

```
feet=int(input("Enter fet:"))

opt=int(input("enter choice 1:inches 2:yards 3:miles 4:millimeters 5.centimeter 6.meters
7.kilometers"))

if opt==1:
    print(round(feet*12,3))

elif opt==2:
    print(round(feet*0.333,3))

elif opt==3:
    print(round(feet*0.000189,3))

elif opt==4:
    print(round(feet*304.8,3))

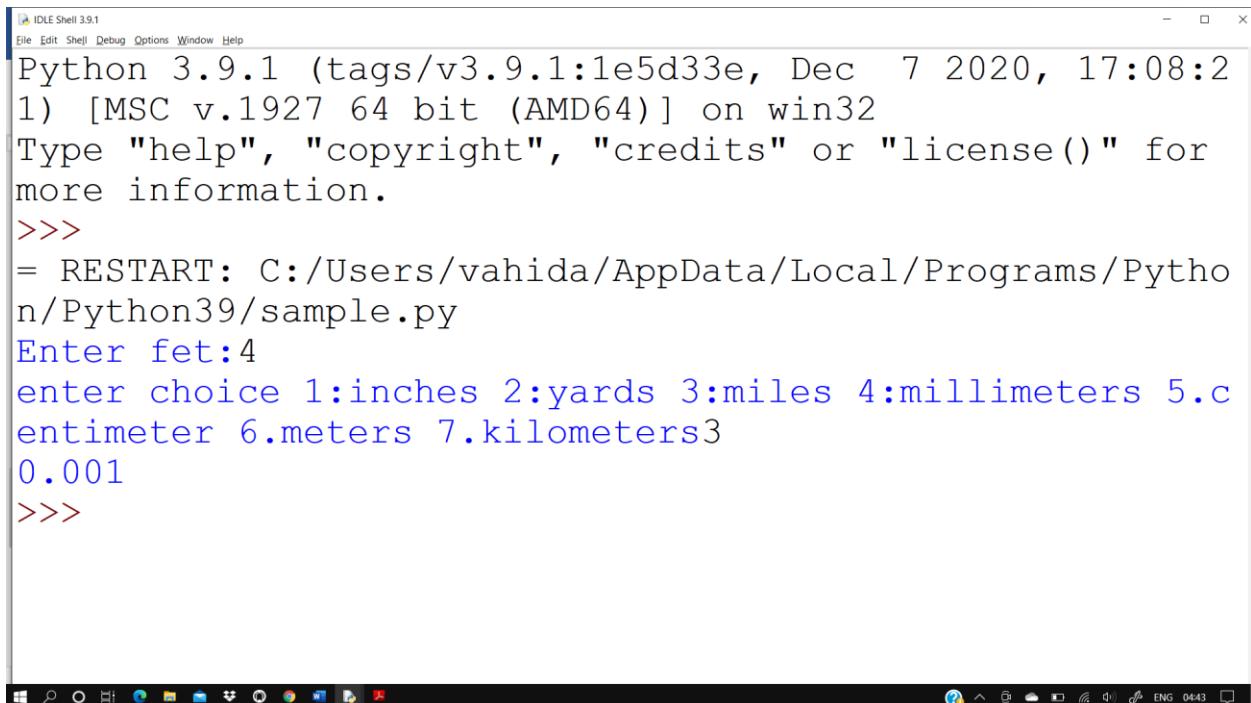
elif opt==5:
    print(round(feet*30.48,3))

elif opt==6:
    print(round(feet*0.305,3))
```

PYTHON PROGRAMMING LAB MANUAL

```
elif opt==7:  
    print(round(feet*0.000305,3))
```

OUTPUT:



The screenshot shows a Windows desktop with a Python script running in the IDLE Shell. The window title is "IDLE Shell 3.9.1". The shell displays the Python version and build information, followed by a help message. It then shows the script being run, prompting for a value, and displaying the converted result.

```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2  
1) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for  
more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho  
n/Python39/sample.py  
Enter fet:4  
enter choice 1:inches 2:yards 3:miles 4:millimeters 5.c  
entimeter 6.meters 7.kilometers3  
0.001  
>>>
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-16

AIM:

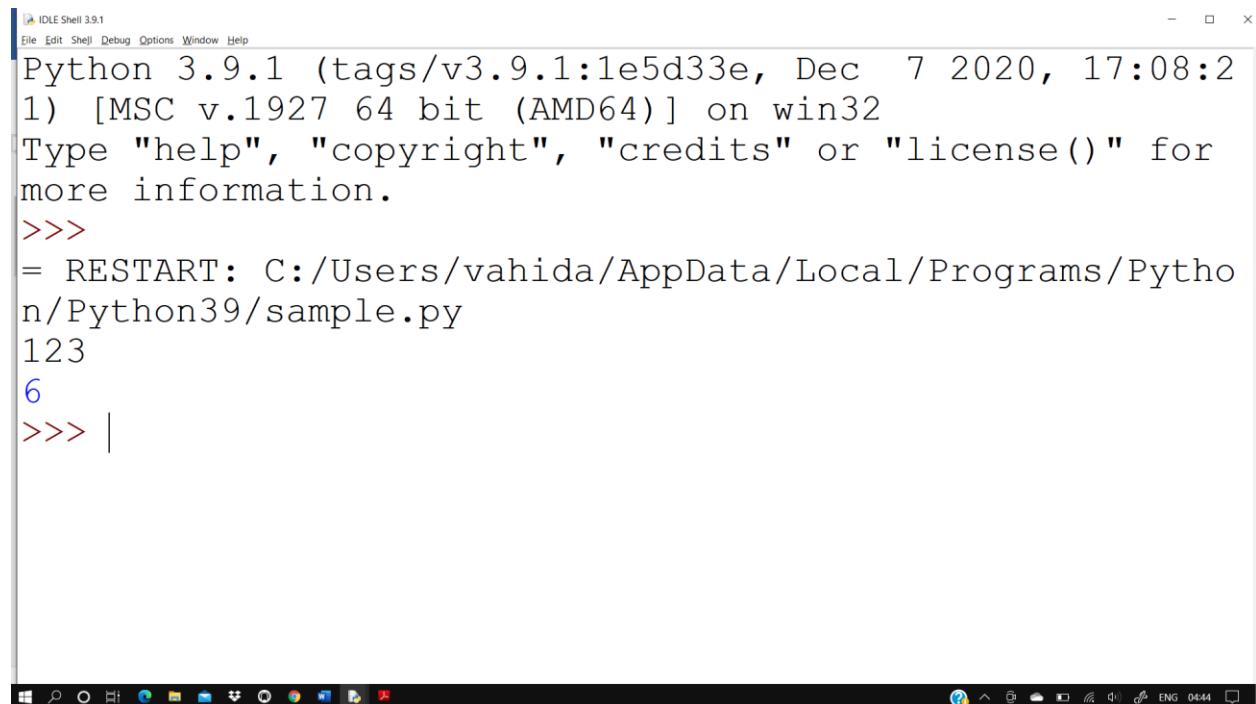
Write a function called sum_digits that is given an integer num and returns the sum of the digits of num.

PROGRAM:

```
def sum_digits(num):
    l=[int(i) for i in str(num)]
    return sum(l)

print(sum_digits(input()))
```

OUTPUT:



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.

>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
123
6
>>> |
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-17

AIM:

Write a function called `first_diff` that is given two strings and returns the first location in which the strings differ. If the strings are identical, it should return -1.

PROGRAM:

```
def first_diff (s1,s2):  
    if s1==s2:  
        return -1  
    else:  
        for i in s1:  
            for j in s2:  
                if s1.index(i)==s2.index(j):  
                    if i!=j:  
                        ind=s1.index(i)  
                        return ind+1  
print(first_diff (input(),input()))
```

OUTPUT:

```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
india
endia
1
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
python
pyton
4
>>> |
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-18

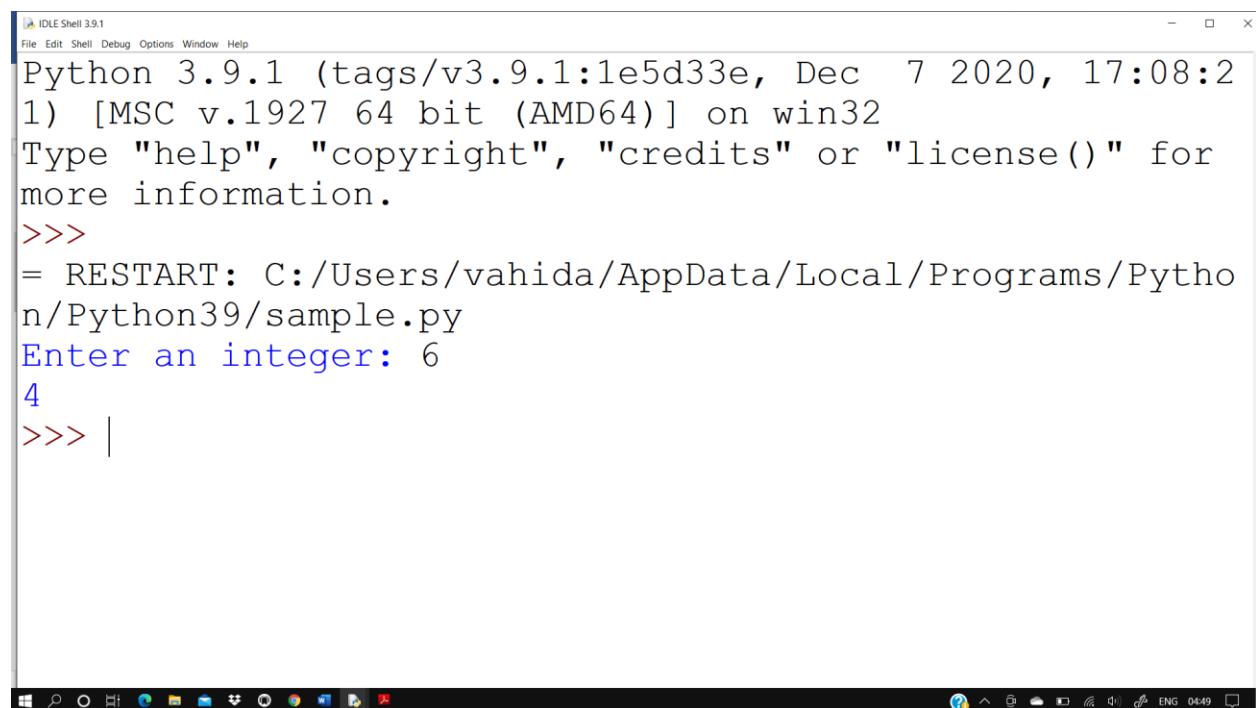
AIM:

Write a function called number_of_factors that takes an integer and returns how many factors the number has.

PROGRAM:

```
def number_of_factors(num):  
  
    factors=[i for i in range(1,num+1) if num%i==0]  
  
    return len(factors)  
  
num=int(input("Enter an integer: "))  
  
print(number_of_factors(num))
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2  
1) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for  
more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho  
n/Python39/sample.py  
Enter an integer: 6  
4  
>>> |
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-19

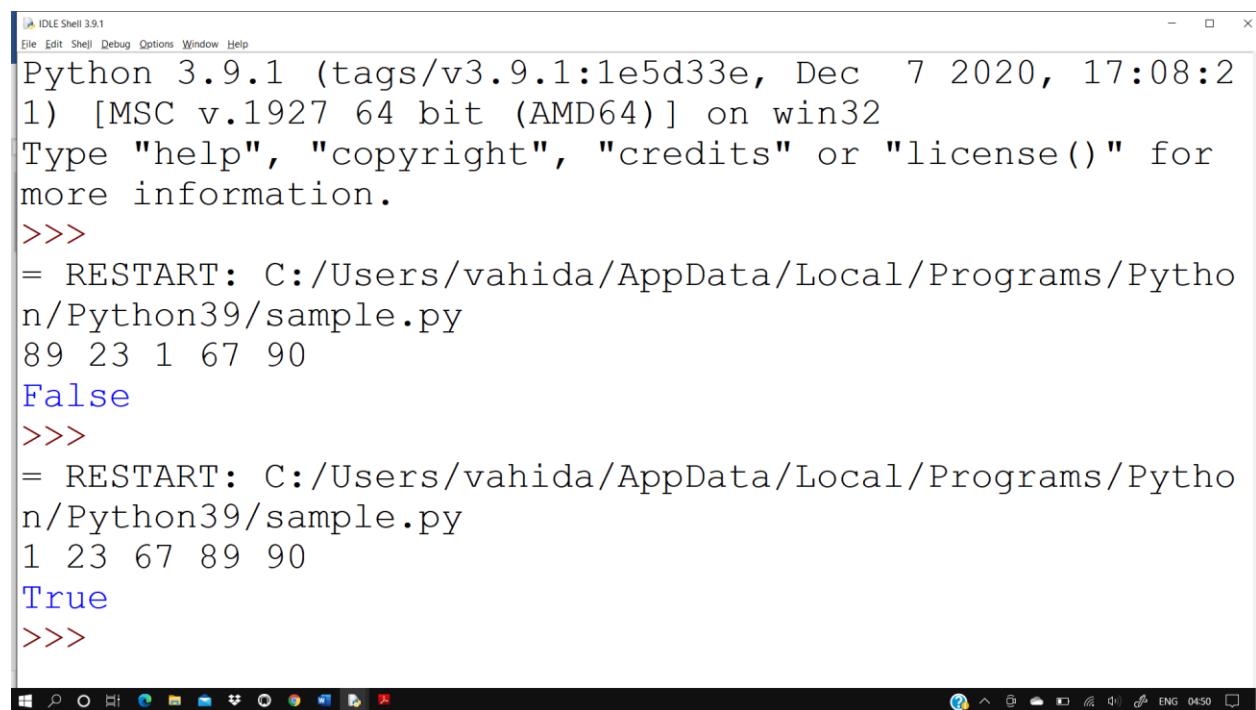
AIM:

write a function called `is_sorted` that is given a list and returns True if the list is sorted and False otherwise

PROGRAM:

```
def is_sorted(l):
    r=sorted(l)
    if r==l:
        return True
    else:
        return False
l=[int(i) for i in input().split()]
print(is_sorted(l))
```

OUTPUT:



The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.9.1". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main window displays the following Python session:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.

>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
89 23 1 67 90
False
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
1 23 67 89 90
True
>>>
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as "ENG 04:50".

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-20

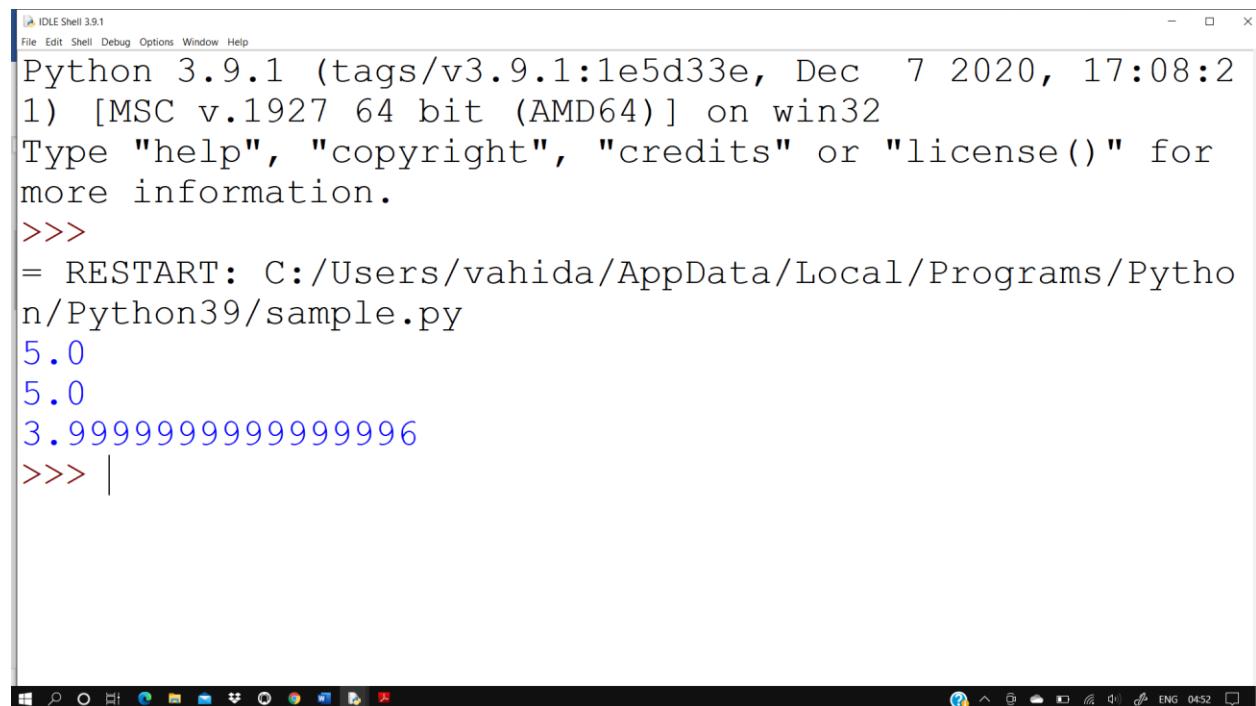
AIM:

write a function root that is given x and an integer n and returns x power 1/n.In the function definition, set the default value of n to 2

PROGRAM:

```
def root(x,n=2):  
    return (x)**(1/n)  
  
print(root(25))  
  
print(root(125,3))  
  
print(root(64,3))
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2  
1) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for  
more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho  
n/Python39/sample.py  
5.0  
5.0  
3.999999999999996  
>>> |
```

EXERCISE-21

AIM:

Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list.

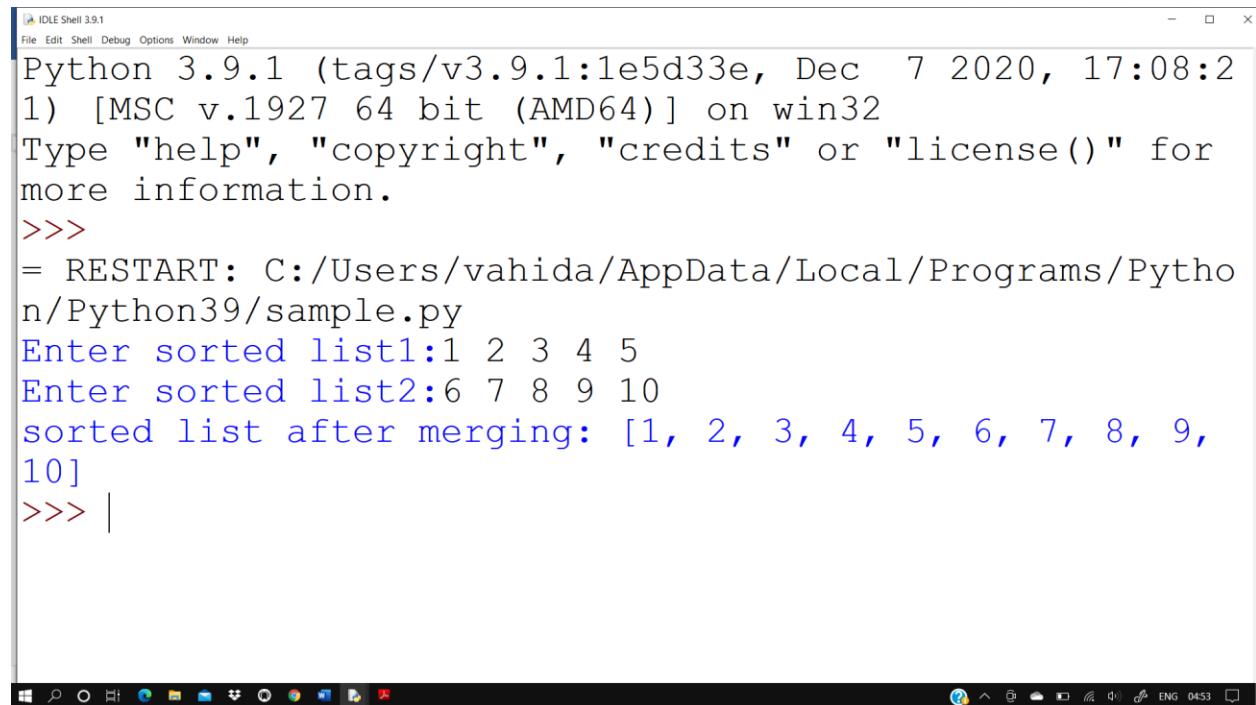
(a) Do this using the sort method. (b) Do this without using the sort method.

PROGRAM:

(a) Do this using the sort method.

```
def merge(s1,s2):
    s=s1+s2
    s.sort()
    return s
a=list(map(int,input("Enter sorted list1:").split()))
b=list(map(int,input("Enter sorted list2:").split()))
x=merge(a,b)
print("sorted list after merging:",x)
```

OUTPUT:



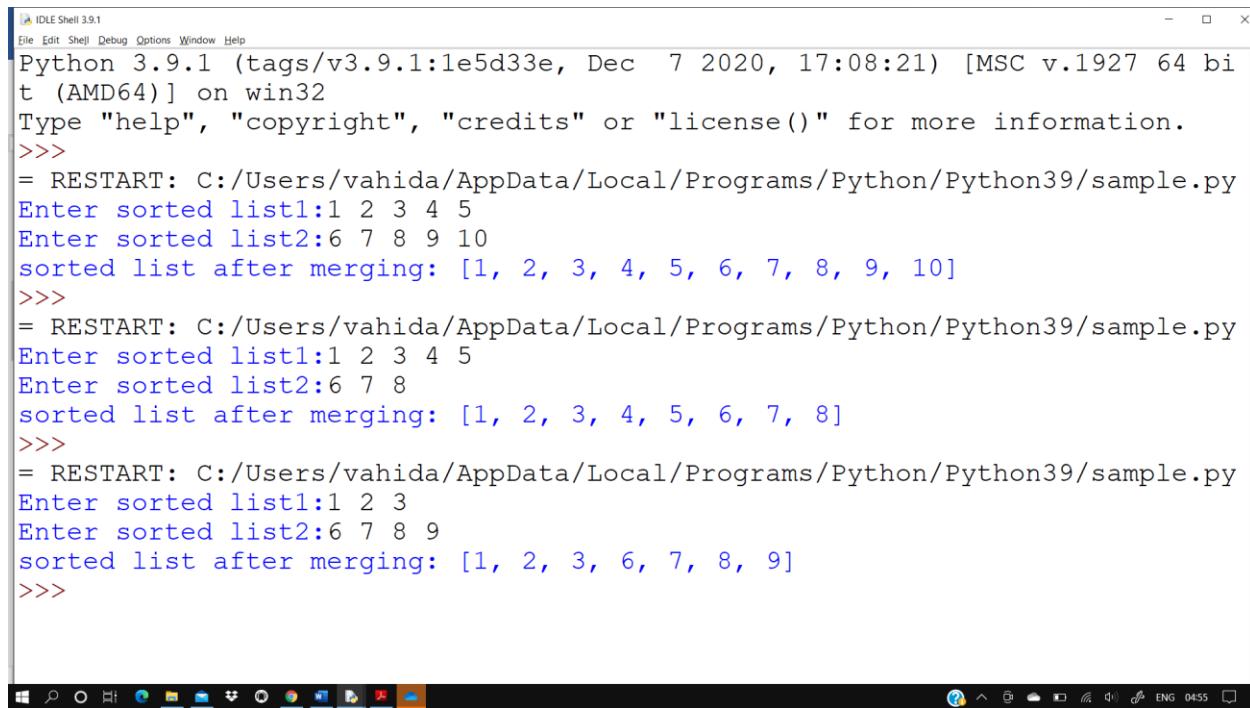
```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:2
1) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Pytho
n/Python39/sample.py
Enter sorted list1:1 2 3 4 5
Enter sorted list2:6 7 8 9 10
sorted list after merging: [1, 2, 3, 4, 5, 6, 7, 8, 9,
10]
>>> |
```

(b) Do this without using the sort method.

```
def merge(l1,l2):
    s1=len(l1)
    s2=len(l2)
    l=[]
    i,j=0,0
    while i<s1 and j<s2:
        if l1[i]<l2[j]:
            l.append(l1[i])
            i+=1
        else:
            l.append(l2[j])
            j+=1
    return l+l1[i:]+l2[j:]

a=list(map(int,input("Enter sorted list1:").split()))
b=list(map(int,input("Enter sorted list2:").split()))
print("sorted list after merging:",merge(a,b))
```

OUTPUT:



The screenshot shows the Python IDLE Shell interface with three separate runs of the merge function. Each run prompts for two sorted lists and prints the resulting merged list. The first run uses lists [1, 2, 3, 4, 5] and [6, 7, 8, 9, 10], the second uses [1, 2, 3, 4, 5] and [6, 7, 8], and the third uses [1, 2, 3] and [6, 7, 8, 9]. The shell window has a title bar 'IDLE Shell 3.9.1' and a menu bar with File, Edit, Shell, Debug, Options, Window, Help.

```
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter sorted list1:1 2 3 4 5
Enter sorted list2:6 7 8 9 10
sorted list after merging: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter sorted list1:1 2 3 4 5
Enter sorted list2:6 7 8
sorted list after merging: [1, 2, 3, 4, 5, 6, 7, 8]
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter sorted list1:1 2 3
Enter sorted list2:6 7 8 9
sorted list after merging: [1, 2, 3, 6, 7, 8, 9]
```

EXERCISE-22

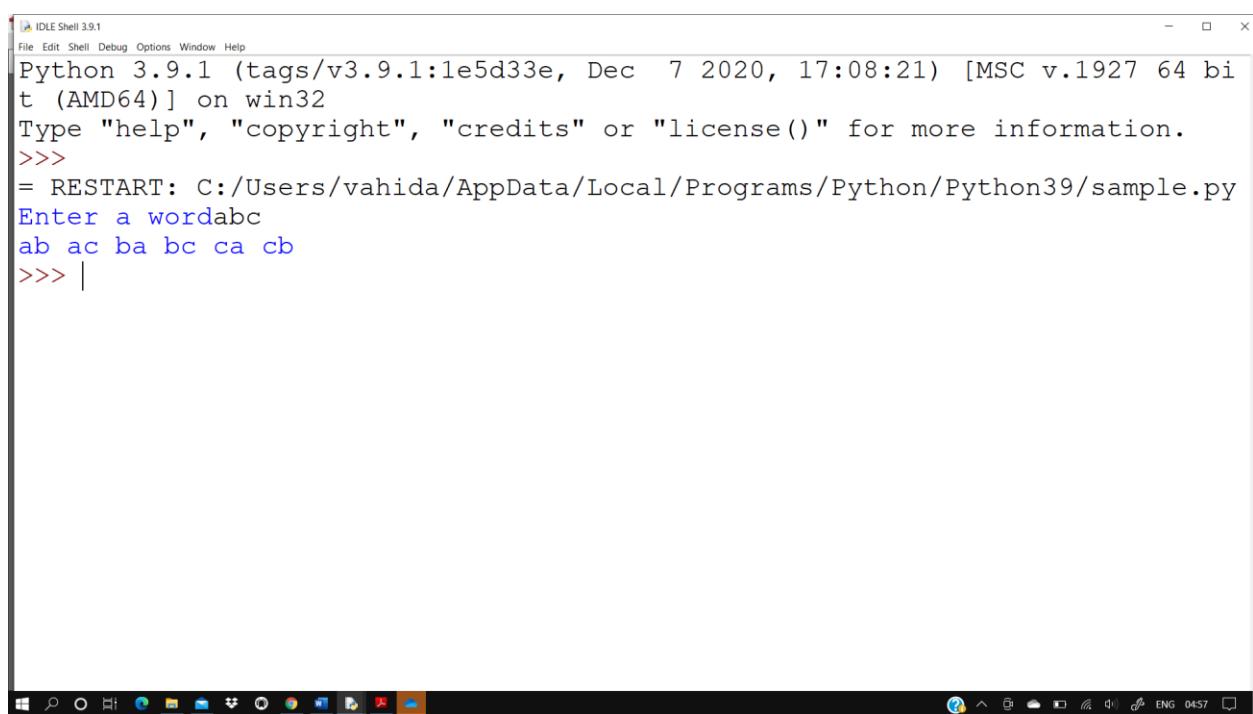
AIM:

Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user's word.

PROGRAM:

```
from itertools import permutations  
  
s=input("Enter a word")  
  
for i in range(2,len(s)):  
  
    for p in permutations(s,i):  
  
        print("".join(p),end=' ')
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bi  
t (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py  
Enter a wordabc  
ab ac ba bc ca cb  
>>> |
```

EXERCISE-23

AIM:

Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons.

PROGRAM:

```
file=open(input("Enter filename:"),'r')

lines=file.readlines()

for line in range(len(lines)):

    if line==len(lines)-1:

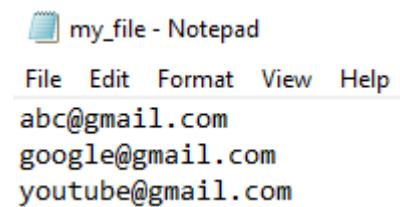
        print('{ }'.format(lines[line].strip()))

    else:

        print('{ }'.format(lines[line].strip()),end=";")
```

OUTPUT:

Enter file name: my_file.txt
abc@gmail.com;google@gmail.com;youtube@gmail.com



EXERCISE-24

AIM:

Write a program that reads a list of temperatures from a file called temps.txt, converts those temperatures to Fahrenheit, and writes the results to a file called ftemps.txt.

PROGRAM:

```
file1=open('temps.txt','r')
lines=file1.readlines()
file2=open('ftemps.txt','w')
for i in range(len(lines)):
    c=lines[i].strip()
    f=round((float(c)*1.8)+32,2)
    file2.write(str(f)+"\n")
file2.close()
```

INPUT FILE:

 ftemp - Notepad
File Edit Format View Help
114.26
112.1
112.64
109.94
108.14
111.38

OUTPUT FILE:

 ftemp - Notepad
File Edit Format View Help
114.26
112.1
112.64
109.94
108.14
111.38

EXERCISE-25

AIM:

Write a class called Product. The class should have fields called name, amount, and price, holding the product's name, the number of items of that product in stock, and the regular price of the product. There should be a method get_price that receives the number of items to be bought and returns the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called make_purchase that receives the number of items to be bought and decreases amount by that much.

PROGRAM:

class product:

```
def __init__(self,name,items,price):
    self.name=name;
    self.items=items
    self.price=price

def getprice(self,n):
    if n<10:
        print("Regular price is charged for your orders")
        cost=n*self.price
    print("If you place above 9 items you get 10% discount")
    print("If you place above 99 items you get 20% discount")

    elif n>=10 and n<100:
```

PYTHON PROGRAMMING LAB MANUAL

```
print("10% dicount is applied for you orders")

cost=n*self.price

discount=(cost*10)/100

finalcost=cost-discount

print("Actual Cost: ",cost)

print("10% Discount: ",finalcost)

print("Cost after 10% discount: ",discount)

print("If you place above 99 items you get 20% discount")

else:

    print("20% dicount is applied for you orders")

    cost=n*self.price

    discount=(cost*20)/100

    finalcost=cost-discount

    print("Actual Cost: ",cost)

    print("20% Discount: ",discount)

    print("Cost after 20% discount: ",finalcost)

def my_purchase(self,n):

    if n<10:

        print("Regular price is charged for you orders")

        cost=n*self.price

        print("Final cost:",cost)

    elif (n>=10) and (n<100):

        print("10% dicount is applied for you orders")
```

PYTHON PROGRAMMING LAB MANUAL

```
cost=n*self.price  
discount=(cost*10)/100  
finalcost=cost-discount  
print("Actual Cost: ",cost)  
print("10% Discount: ",discount)  
print("Final Cost after 10% discount: ",finalcost)  
  
else:  
    print("20% dicount is applied for you orders")  
    cost=n*self.price  
    discount=(cost*20)/100  
    finalcost=cost-discount  
    print("Actual Cost: ",cost)  
    print("20% Discount: ",discount)  
    print("Final Cost after 20% discount: ",finalcost)  
  
p=product("PEN",200,5)  
n=int(input("Enter Number of pens you want to buy:"))  
p.getprice(n)  
n=int(input("Enter Number of pens you want to buy:"))  
p.my_purchase(n)
```

PYTHON PROGRAMMING LAB MANUAL

OUTPUT:

The screenshot shows a Windows desktop environment with a Python IDLE Shell window open. The window title is "IDLE Shell 3.9.1". The code in the shell window is as follows:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter Number of pens you want to buy:56
10% dicount is applied for you orders
Actual Cost: 280
10% Discount: 252.0
Cost after 10% discount: 28.0
If you place above 99 items you get 20% discount
Enter Number of pens you want to buy:|
```

The taskbar at the bottom of the screen shows various application icons and system status indicators.

EXERCISE-26**AIM:**

Write a class called Time whose only field is a time in seconds. It should have a method called convert_to_minutes that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called convert_to_hours that returns a string of hours, minutes, and seconds formatted analogously to the previous method.

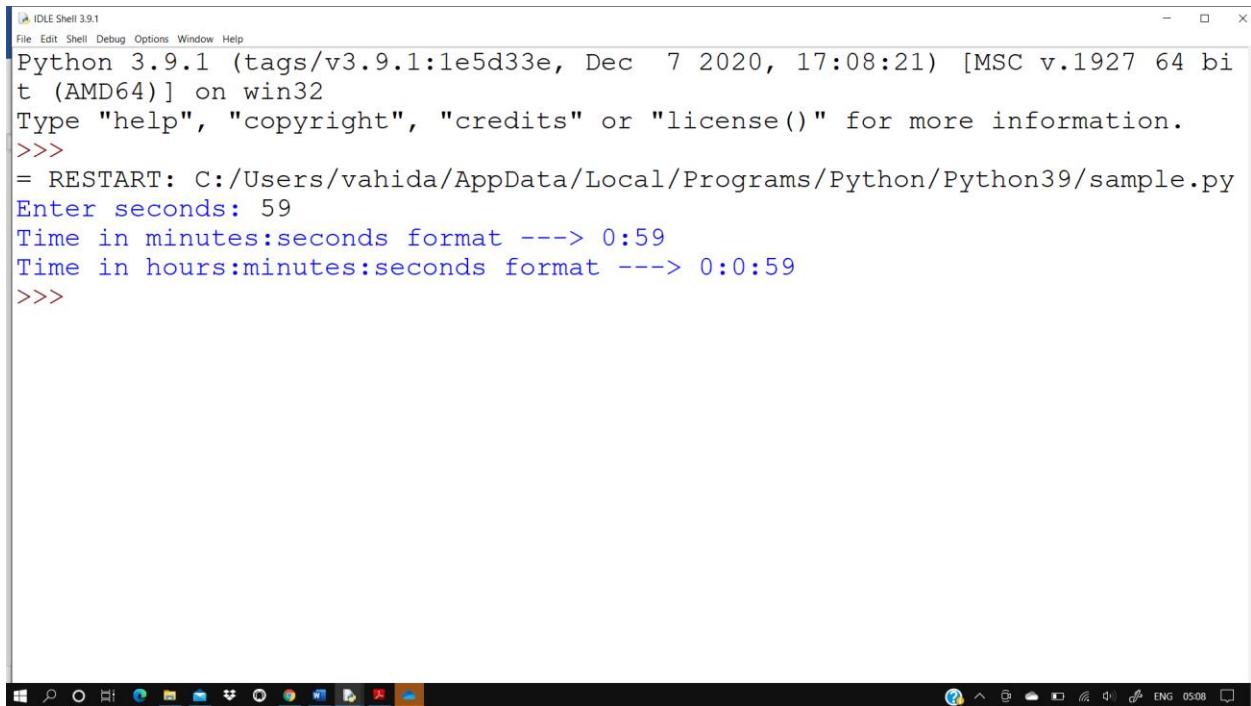
PROGRAM:

class Time:

```
def __init__(self,sec):
    self.sec=sec
def convert_to_minutes(self):
    n=self.sec
    minutes=n//60
    seconds=n%60
    return(str(minutes)+":"+str(seconds))
def convert_to_hours(self):
    n=self.sec
    hours=n//3600
    minutes=(n//60)%60
    seconds=n%60
    return(str(hours)+":"+str(minutes)+":"+str(seconds))
a=int(input("Enter seconds: "))
c=Time(a)
print("Time in minutes:seconds format --->",c.convert_to_minutes())
print("Time in hours:minutes:seconds format --->",c.convert_to_hours())
```

PYTHON PROGRAMMING LAB MANUAL

OUTPUT:



The screenshot shows a Windows desktop environment with a Python IDLE Shell window open. The window title is "IDLE Shell 3.9.1". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the following Python session:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter seconds: 59
Time in minutes:seconds format ---> 0:59
Time in hours:minutes:seconds format ---> 0:0:59
>>>
```

The taskbar at the bottom shows various pinned icons and the system clock indicating 05:08.

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-27

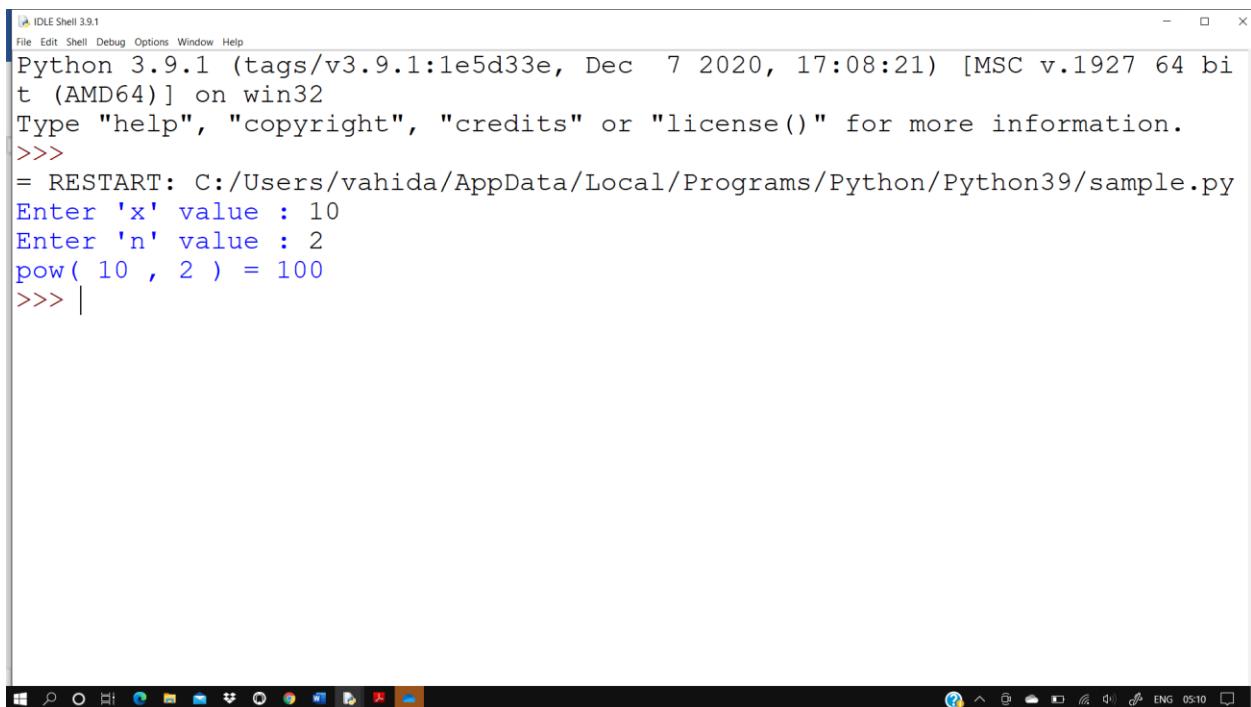
AIM:

Write a Python class to implement $\text{pow}(x, n)$.

PROGRAM:

```
class power:  
    def pow(self,x,n):  
        print("pow(",x,") =",x**n)  
  
p=power()  
  
x=int(input("Enter 'x' value : "))  
  
n=int(input("Enter 'n' value : "))  
  
p.pow(x,n)
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py  
Enter 'x' value : 10  
Enter 'n' value : 2  
pow( 10 , 2 ) = 100  
>>> |
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-28

AIM:

Write a Python class to reverse a string word by word.

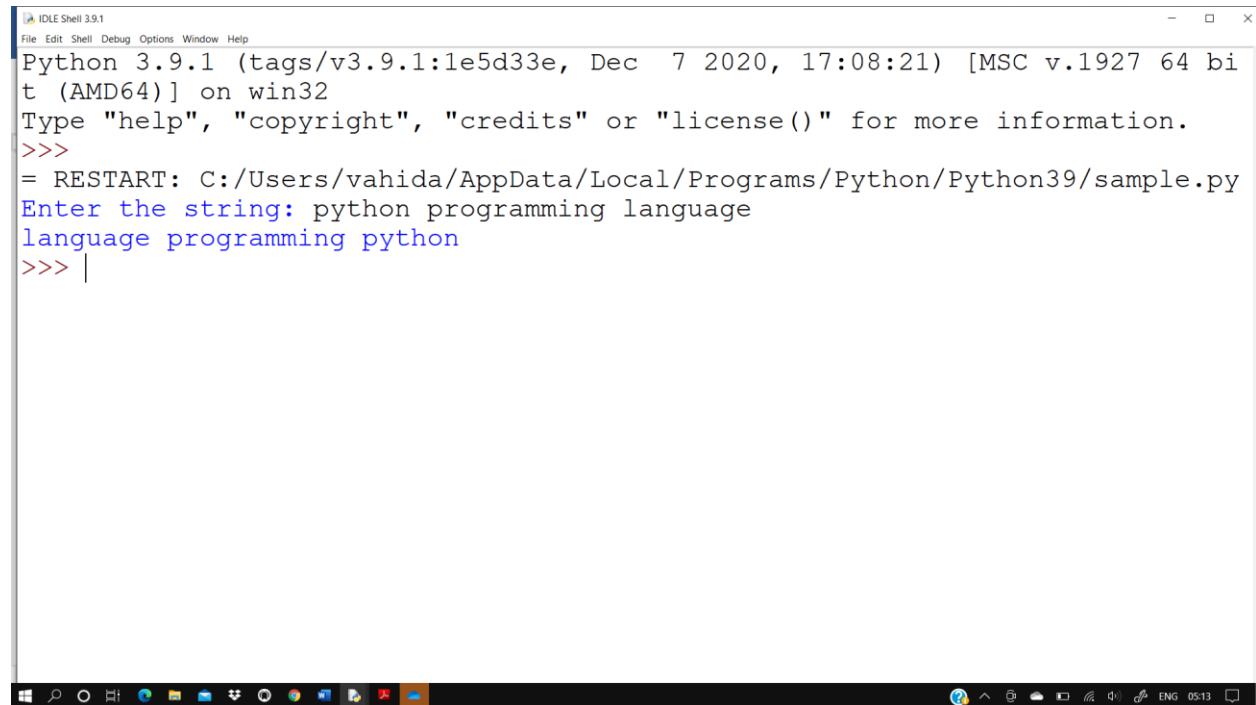
PROGRAM:

```
class reverse:
```

```
    def rev_sentence(self,sentence):
        words = sentence.split(' ')
        reverse_sentence = ''.join(reversed(words))
        print(reverse_sentence)

c=reverse()
c.rev_sentence(input("Enter the string: "))
```

OUTPUT:



The screenshot shows the Python IDLE Shell 3.9.1 interface. The window title is "IDLE Shell 3.9.1". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main console area displays the following output:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter the string: python programming language
language programming python
>>> |
```

The taskbar at the bottom of the screen shows various application icons, including Microsoft Edge, File Explorer, and other system icons. The system tray indicates the date and time as 05:13.

EXERCISE-29

AIM:

Write a program to demonstrate Try/except/else.

PROGRAM:

try:

```
a=int(input("Enter 'a' value:"))
```

```
b=int(input("Enter 'b' value:"))
```

```
c=a/b
```

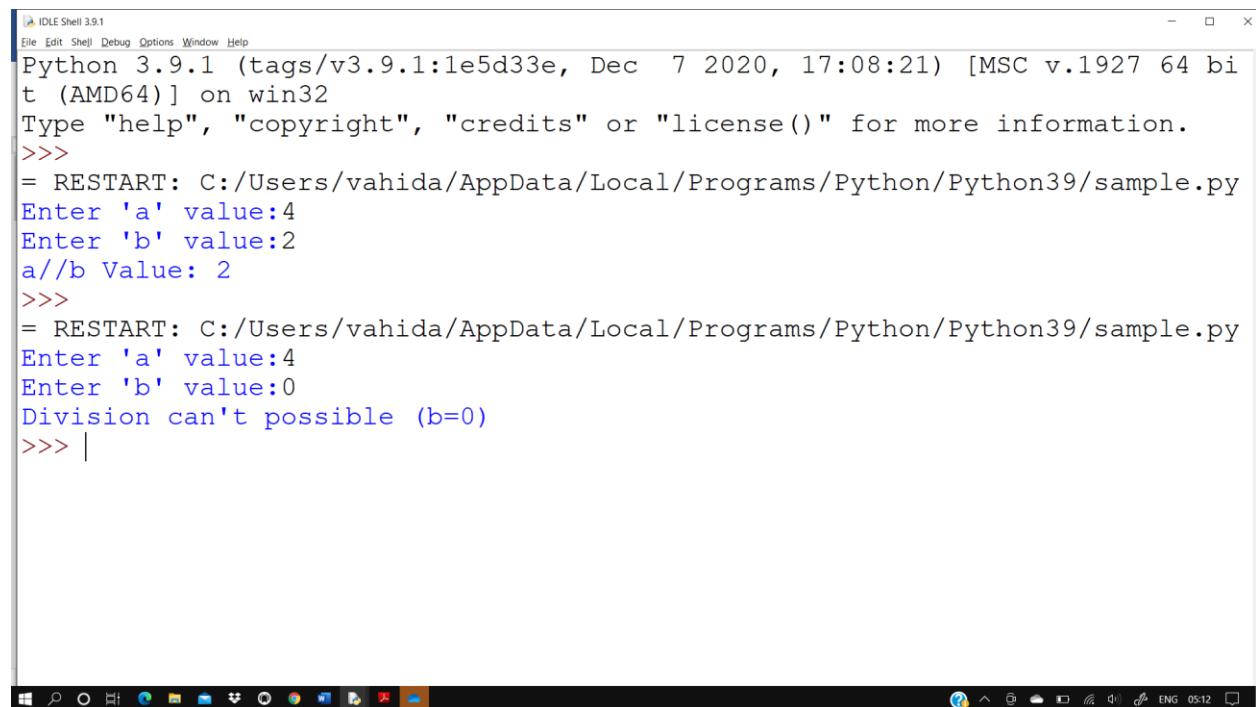
except ZeroDivisionError:

```
    print("Division can't possible (b=0)")
```

else:

```
    print("a/b Value:",c)
```

OUTPUT:



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter 'a' value:4
Enter 'b' value:2
a//b Value: 2
>>>
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py
Enter 'a' value:4
Enter 'b' value:0
Division can't possible (b=0)
>>> |
```

PYTHON PROGRAMMING LAB MANUAL

EXERCISE-30

AIM:

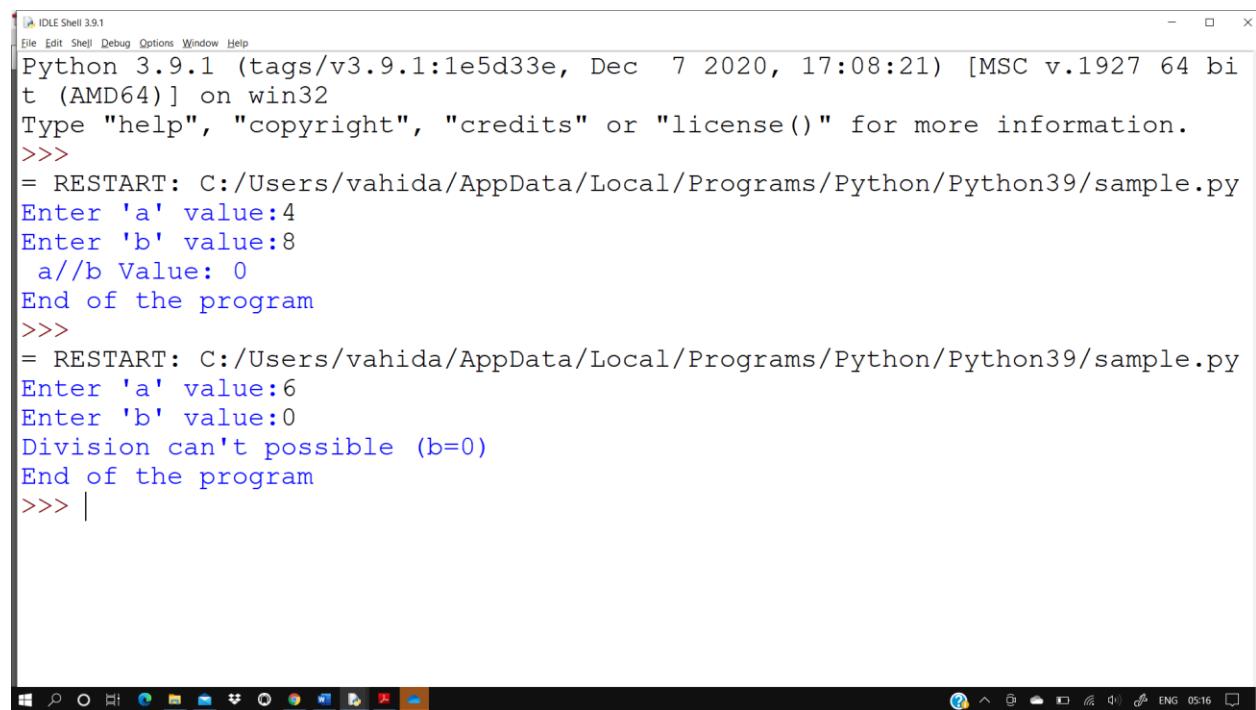
Write a program to demonstrate try/finally and with/as.

PROGRAM:

Program for try/finally:

```
try:  
    a=int(input("Enter 'a' value:"))  
    b=int(input("Enter 'b' value:"))  
    c=a//b  
  
except ZeroDivisionError:  
    print("Division can't possible (b=0)")  
  
else:  
    print(" a/b Value:",c)  
  
finally:  
    print("End of the program")
```

OUTPUT:



```
IDLE Shell 3.9.1  
File Edit Shell Debug Options Window Help  
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bi  
t (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py  
Enter 'a' value:4  
Enter 'b' value:8  
 a//b Value: 0  
End of the program  
>>>  
= RESTART: C:/Users/vahida/AppData/Local/Programs/Python/Python39/sample.py  
Enter 'a' value:6  
Enter 'b' value:0  
Division can't possible (b=0)  
End of the program  
>>> |
```

program for with/as:

```
file=open('file1.txt','w')
try:
    file.write('hello friends how are you')
finally:
    file.close()

with open('C:\Users\ACET\Desktop\text.txt','w') as file:
    file.write('hello friends how are you')
```

OUTPUT:

file1.txt and file2.txt created in the current directory.

