LAB CYCLE -1

Experiment No :1

Date :30/09/2024

Aim :

Write a program that prompts the user to enter his first name and last name and then displays a message “Greetings!!! First name Last name”.

Pseudocode :

```
1. Read first name.
2. Read last name.
3. Print “Greetings!!!First name Last name”
```

Source Code :

```
first_name=input("Enter your first name:")
last_name=input("Enter your last name:")
print(f"Greetings!!! {first_name} {last_name}")
```

Output :

```
Enter your first name: Nafia
Enter your last name: V
Greetings!!! Nafia V
```

Result :

The program is successfully executed and the output is verified.

Experiment No :2

Date :30/09/2024

Aim :

Write a program to demonstrate different number data types in python.

Pseudocode :

```
1. Assign 3 type of values to corresponding variables.
2. Print datatype of each variables using type() function.
```

Method :

Function	Description	Syntax
type()	returns the type of the specified object	type(object, bases, dict)

Source Code :

```
int_num=10
print(f"Integer: {int_num},Type:{type(int_num)}")
float=10.5
print(f"Float: {float},Type:{type(float)}")
complex_num=3+5j
print(f"Complex Number: {complex_num},Type:{type(complex_num)}")
```

Output :

```
Integer: 10 ,Type: <class 'int'>
Float: 10.5 ,Type: <class ‘float’>
Complex Number: (3+5j) ,Type: <class 'complex'>
```

Result :

The program is successfully executed and the output is verified.

Experiment No :3

Date :30/09/2024

Aim :

Write a program to calculate the area of a circle by reading inputs from the user.

Pseudocode :

```
1. Read the radius r of the circle.
2. Area=π*r*r
3. Print Area
```

Source Code :

```
radius=float(input(“Enter the radius :”))
area=3.14*radius*radius
print(f“The area of circle with radius {radius} is {area}”)
```

Output :

```
Enter the radius:4
The area of circle with radius 4.0 is 50.24
```

Result :

The program is successfully executed and the output is verified.

Experiment No :4

Date :30/09/2024

Aim :

Write a program to calculate the salary of an employee given his basic pay (to be entered by the user) . HRA = 10 percent of the basic pay, TA = 5 percent of the basic pay.

Pseudocode:

```
1. Read the basic pay of the employee.
2. Calculate HRA,TA and Salary
3. Print Salary
```

Source Code :

```
basic_pay=float(input("Enter the basic pay:"))
hra=0.10*basic_pay
ta=0.05*basic_pay
total=hra+ta+basic_pay
print(f"Basic pay:{basic_pay}\nHRA:{hra}\nTA:{ta}\nTotal:{total}")
```

Output :

```
Enter the basic pay:3000
Basic pay= 3000.0
HRA:300.0
TA:150.0
Total=3450.0
```

Result :

The program is successfully executed and the output is verified.

Experiment No :5

Date :30/09/2024

Aim :

Write a Python program to perform arithmetic operations on two integer numbers.

Pseudocode :

```
1. Read two numbers num1 and num2.
2. Print num1 + num2
3. Print num1 – num2
4. Print num1 * num2
5. Print num1 / num2
6. Print num1 % num2
7. Print num1 ** num2
8. Print num1 // num2
```

Source Code :

```
num1=int(input("Enter the first number: "))
num2=int(input("Enter the second number: "))
sum=num1+num2
difference=num1- num2
product=num1* num2
division=num1/num2
modulus=num1%num2
exponent=num1**num2
floordiv=num1//num2
print(f"Sum: {num1}+{num2}={sum}")
print(f"Difference: {num1}-{num2}={difference}")
print(f"Product: {num1}*{num2}={product}")
print(f"Division: {num1}/{num2}={division}")
print(f"Modulus: {num1}%{num2}={modulus}")
print(f"Exponent: {num1}**{num2}={exponent}")
print(f’Floor division: {num1}//{num2}={floordiv}”)
```

Output :

```
Enter the first number: 5
Enter the second number: 7
Sum: 5 + 7 = 12
Difference: 5 - 7 = -2
Product : 5 * 7 = 35
Division: 5 / 7 = 0.714857
Modulus: 5 % 7 = 5
Exponent: 5**7=78125
Floor division: 5//7=0
```

Result :

The program is successfully executed and the output is verified.

Experiment No :6

Date : 30/09/2024

Aim :

Write a Python program to get a string which is n (non-negative integer) copies of a given string.

Pseudocode :

```
1. Read string .
2. Read number of repetition n.
3. Print s*n. //print string as the specified number of times
```

Source Code :

```
string=input("Enter the string:")
n=int(input("Enter the number of copies:"))
c=string*n
print(“copies of string are:”,c )
```

Output :

```
Enter the string:python
Enter the number of copies:3
Copies of string are:pythonpythonpython
```

Result :

The program is successfully executed and the output is verified.

Experiment No :7

Date : 30/09/2024

Aim :

Program to accept an integer n and compute n+nn+nnn.

[Hint : n = 5, then compute 5 + 55 + 555]

Pseudocode:

```
1. Read a number n
2. result= n+ nn+nnn
3. Print result.
```

Source Code :

```
n=input(“Enter the number:”)
result=int(n)+int(n*2)+int(n*3)
print(f”{n}+{n}{n}+{n}{n}{n}={result}”)
```

Output :

```
Enter the number:5
5+ 55+ 555=615
```

Result :

The program is successfully executed and the output is verified.

Experiment No :8

Date :07/10/2024

Aim :

Find biggest of three numbers entered.

Pseudocode :

```
1. Read numbers num1,num2,num3.
2. if num1>num2 and num1>num3 then
         Print num1 is greatest.
   else if num2>num3 and num2>num1 then
         Print num2 is greatest.
   else
         Print num3 is greatest.
   end if
```

Source Code :

```
num1=int(input("Enter the first number:"))
num2=int(input("Enter the second number:"))
num3=int(input("Enter the third number:"))
if num1>num2 and num1>num3:
    print(f"{num1} is greatest")
elif num2>num3 and num2>num1:
    print(f"{num2} is greatest")
else:
    print(f"{num3} is greatest")’
```

Output :

```
Enter the first number:11
Enter the second number:45
Enter the third number:67
67 is greatest
```

Result :

The program is successfully executed and the output is verified.

Experiment No :9

Date : 07/10/2024

Aim :

Program to determine whether a year is a leap year or not

Pseudocode :

```
1. Read year.
2. if year mod 4!=0 then
         print " not a leap year"
   else if year mod 100!=0 then
         print "is a leap year"
   else if year mod 400 == 0 then
         print "is a leap year"
   else
         print "not a leap year"
   end if
```

Source Code :

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

Output :

```
Enter  a year:2024
It is  a leap year
Enter a year:2021
not a leap year
```

Result :

The program is successfully executed and the output is verified.

Experiment No :10

Date :07/10/2024

Aim :

Write a python program to determine the rate of Entry-ticket in a trade fair based on age as follows:

Age	Rate
<10	7
>=10 and <60	10
>=60	5

Pseudocode :

```
1. Read Age.
2. if age<10 then
            print Your ticket rate for trade fair is:7
   else if 10<=age<60 then
            print Your ticket rate for trade fair is:10
   else if age>=60 then
            print Your ticket rate for trade fair is:5
   else
            print Enter proper age
   end if
```

Source Code :

```
age=int(input("Enter your age : "))
if age<10:
    print("Your ticket rate for trade fair is: 7")
elif 10<=age<60:
    print("Your ticket rate for trade fair is: 10")
```

```
elif age>=60:
    print("Your ticket rate for trade fair is: 5")
 else:
     print("Enter a proper age")
```

Output :

```
Enter your age : 5
Your ticket rate for trade fair is : 7
Enter your age : 65
Your ticket rate for trade fair is : 5
Enter your age : 20
Your ticket rate for trade is : 10
```

Result :

The program is successfully executed and the output is verified.

Experiment No :11

Date : 07/10/2024

Aim :

Write a Python program to solve a quadratic equation.

Pseudocode :

```
1. Read coefficients a,b,c.
2. Print quadratic equation in the form ax^2+bx+c=0
3. Set discriminant=b**2-4*a*c
4. if discriminant>0 then
        Set root1=b+sqrt((discriminant)/(2*a))
        Set root2=b-sqrt((discriminant)/(2*a))
        Print “The roots are real and different”
        Print the roots
   else if discriminant== 0 then
        Set root=-b/(2*a)
        Print “The roots are real and equal”
        Print the root
   else if discriminant<0:
        realpart=-b/(2*a)
        imaginarypart=math.sqrt(-discriminant)/(2*a)
        Print “The roots are complex”
        Print the roots
   else:
        print”The equation has no real roots!”
   end if
```

Method :

Function	Description	Syntax
sqrt()	returns the squareroot of a number	math.sqrt(x)

Source Code :

```
import math
a=float(input("Enter the coefficient x^2: "))
b=float(input("Enter the coefficient x: "))
c=float(input("Enter the constant: "))
print(f’Quadratic equation:{a}x^2+{b}x+{c}=0\n”)
discriminant=b**2 - 4*a*c
if discriminant>0:
    root1=(b+math.sqrt(discriminant))/(2*a)
    root2=(b-math.sqrt(discriminant))/(2*a)
    print(f”The roots are real and different”)
    print(f”The roots are {root1:2f}and{root2:2f}\n”)
elif discriminant==0:
    root=b/(2*a)
    print(f"The roots are real and equal")
    print(f”The root is {root:2f}\n”)
elif discriminant<0:
    realpart=b/(2*a)
    imaginarypart=math.sqrt(-discriminant)/(2*a)
    print(f”The roots are complex”)
    print(f"The roots are {realpart:2f}+{imaginarypart:2f}and {realpart:2f}-
{imaginary part:2f}\0”)
else:
    print(“The equation has no real roots!\n”)
```

Output :

```
Enter the coefficient x^2: 1
Enter the coefficient x: 1
Enter the constant: 1
Quadratic equation:1.0x^2+1.0x+1.0=0
The roots are complex
The roots are 0.500000+0.866025 and 0.500000-0.8666025

Enter the coefficient x^2: 1
Enter the coefficient x: 2
Enter the constant: 1
Quadratic equation:1.0x^2+2.0x+1.0=0
The roots are real and equal
The root is 1.00000

Enter the coefficient x^2: 1
Enter the coefficient x: 4
Enter the constant: 3
Quadratic equation:1.0x^2+4.0x+3.0=0
The roots are real and different
The roots are 3.00000 and 1.00000
```

Result :

The program is successfully executed and the output is verified.

LAB CYCLE -2

Experiment No :1

Date : 07/10/2024

Aim :

Create a string from the given string where the first and last character are exchanged.

Eg: Python ⇒ nythoP

Pseudocode :

```
1.Read string str.
2.Print rearranged string sliced using index.
  newstring= str[-1] + str[1:-1] + str[0]
3.Print newstring.
```

Source Code :

```
str=input("Enter a string:")
newstring =str[-1]+str[1:-1]+str[0]
print(f"newstring:{newstring}")
```

Output :

```
Enter a string: python
newstring:nythop
```

Result :

The program is successfully executed and the output is verified.

Experiment No :2

Date : 07/10/2024

Aim :

Get a string from an input string where all occurences of first character are replaced with $,except the first character. [eg:onion->oni$n]

Pseudocode :

```
1.Read string str.
2.Convert string into lowercase and store in str2
3.Store str[0] into firstchar
4.Replace all occurences of firstchar in substring with '$'+firstchar and store into
  newstr
5.Print newstr
```

Method :

Function	Description	Syntax
lower()	Converts all uppercase characters in a string to lowercase	string.lower()
replace()	Replaces occurences of substring with another string	String.replace(old, new,count)

Source Code :

```
str=input("Enter a string which has reoccurance of first character :")
firstchar=str[0]
newstr=firstchar+str[1: ].replace(firstchar,'$')
print(f"New string is :{newstr}")
```

Output :

```
Enter a string which has reoccurance of first character : pythonp
New string is : python$
```

Result :

The program is successfully executed and the output is verified.

Experiment No :3

Date : 07/10/2024

Aim :

Create a single string separated with space from two strings by swapping the character at position 1.

Eg : str1 = “Hello” str2 =”World” , then create a string str3 = “Hollo Werld” [Hint: use slicing and concatenation]

Pseudocode :

```
1.Set str1=Hello.
2.Set str2= World.
3.Set str3=str1[0]+str2[1]+str1[2:]+ " " +str2[0]+str1[1]+str2[2:]
4.print str3
```

Source Code :

```
str1="Hello"
str2="World"
str3=str1[0]+str2[1]+str1[2: ]+" "+str2[0]+str1[1]+str2[2: ]
print(f"{str3}”)
```

Output :

```
Hollo Werld
```

Result :

The program is successfully executed and the output is verified.

Experiment No :4

Date : 07/10/2024

Aim :

Count the number of characters (character frequency) in a string.

Pseudocode :

```
1.Read string str
2. Convert string into lowercase and store in str2
3.Initialize Empty dictionary called Count
4.for each char in str2 do
          if  char in count then
                 Increment count[char] by 1
          else
                 count[char]=1
          end if
  end for
5.Print count
```

Source Code :

```
str=input("Enter a string:")
count={}
for char in str2:
    if char in count:
        count[char]+=1
    else:
        count[char]=1
print(count)
```

Output :

```
Enter a string: python
{‘p’ : 1, ‘y’ :1, ‘t’ : 1, ‘h’ : 1, ‘o’ : 1, ‘n’ : 1}
```

Result :

The program is successfully executed and the output is verified.

Experiment No :5

Date : 07/10/2024

Aim :

Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’ .

Pseudocode :

```
1.Read string word.
2.if word.endswith(‘ing’) then
        Set str=word+ ‘ly’
  else
        Set str=word+ ‘ing’
  end if
3.Print str.
```

Method:

endswith()	returns True if the string ends with the specified value, otherwise False.	string.endswith(value, start, end)

Source Code :

```
word=input("Enter a string:")
if word.endswith('ing'):
    str=word+'ly'
else:
    str=word+'ing'
print(f"{str}")
```

Output :

```
Enter a string:Read
Reading
Enter a string: Loving
Lovingly
```

Result :

The program is successfully executed and the output is verified.

Experiment No :6

Date : 14/10/2024

Aim :

Store a list of first names. Count the occurrences of ‘a’ within the list.

Pseudocode :

```
1.Store a list of firstnames as names
2. Set frequency=0
3.for each name in names do
    Set frequency=frequency+name.lower().count(‘a’)
  end for
4.Print “Occurrence of ‘a’ in list is”,frequency
```

Method :

Function	Description	Syntax
count()	Returns the Count of the occurrences of a specific substring within a string.	string.count(substring, start,end)

Source Code :

```
names=[“Nafia","Rinu","Aparna","Shahma"]
frequency=0
for name in names:
        frequency+=name.lower().count('a')
print(f"Occurence of 'a' in list is {frequency}")
```

Output :

```
Occurrence of ‘a’ in list is 7
```

Result :

The program is successfully executed and the output is verified.

Experiment No :7

Date : 14/10/2024

Aim :

Write a python program to read two lists color-list1 and color-list2. Print out all colors from color-list1 not contained in color-list2.

Pseudocode :

```
1. Read a set of colors seprated by space as colorlist1
2. Read a set of colors seprated by space as colorlist2
3.  Convert  colorlist1 into a list and stored into color1 using split function
4. Convert  colorlist2 into a list and stored into color2 using split function
5. Initialize list,newlist=[]
6. for each color in colorlist1 do
           if item not in colorlist2 then
                  Append color to newlist
           end if
      end for
7. Print newlist
```

Method :

Function	Description	Syntax
split()	splits a string into a list.	split(separator, maxsplit)

Source Code :

```
colorlist1=input("Enter colors for list 1 seprated by space : ")
colorlist2=input("Enter colors for list 2 seprated by space : ")
color1=colorlist1.split()
color2=colorlist2.split()
```

```
newlist=[color for color in colorlist1 if color not in colorlist2]
print(f"list of colors from list1 and not contained in list2 are : {newlist}")
```

Output:

```
Enter color for list1 seprated by space : green red blue
Enter color for list2 seprated by space : yellow red white
list of colors from list1 and not contained in list2 are : [‘green’, ‘blue’]
```

Result :

The program is successfully executed and the output is verified.

Experiment No :8

Date : 14/10/2024

Aim :

Create a list of colors from comma-separated color names entered by the user. Display first and last colors.

Pseudocode :

```
1. Read a set of colors seprated by comma as colorlst
2. Convert  colorlst into a list and stored into color using split function
3. Print First color of list:color[0]
4. Print Last color of list:color[-1]
```

Source Code :

```
colorlst=input("Enter list of colors seprated by comma : ")
colorlst=colorlst.split(",")
print(f"First color of list:{color[0]}")
print(f"Last color of list:{color[-1]}")
```

Output:

```
Enter list of colors seprated by comma : red,blue,yellow,black
First color of list : red
Last color of list : black
```

Result :

The program is successfully executed and the output is verified.

Experiment No :9

Date : 14/10/2024

Aim :

Write a program to prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

Pseudocode :

```
1. Read list of numbers as n.
2. Initialize list l= []
3. for each i in n do
        Read integer as num
         if num>100 then
                Set num= “over”
        end If
        Append num to listed
   end for
4. Print listed
```

Method :

Function	Description	Syntax
append()	appends an element to the end of the list.	list.append(element)

Source Code:

```
numbers=input("Enter a list of integers separated by space : "))
a=numbers.split()
l=[]
```

```
for i in a:
        l.append(int(i))
        print(“List: “,1)
        m_list=[]
for i in l:
        if i>100:
                m_list.append('over')
        else:
                m_list.append(i)
print(“modified list : “,m_list)
```

Output :

```
Enter a list of integers separated by space : 1 240 3
List: [1]
List: [1, 240]
List: [1, 240, 3]
Modified list : [1, ’over’ , 3]
```

Result :

The program is successfully executed and the output is verified.

Experiment No :10

Date : 14/10/2024

Aim :

From a list of integers, create a list after removing even numbers.

Pseudocode :

```
1. Read list of numbers as n.
2. Initialize list listed = []
3. for each i in n do
        Read integer as num
         if num%2!=0 then
             Append num to listed
        end if
   end for
4. Print listed
```

Source Code :

```
n=int(input("Enter total number of integers to input :"))
listed=[]
for i in range(n):
      num=int(input("Enter Integers : "))
      if int(num)%2!=0:
            listed.append(num)
print(listed)
```

Output :

```
Enter total number of integers to input : 6
Enter integers : 1
Enter integers : 2
Enter integers : 37
Enter integers : 4
Enter integers : 57
Enter integers : 8
[1, 37, 57]
```

Result :

The program is successfully executed and the output is verified.

Experiment No :11

Date : 14/10/2024

Aim :

Accept a list of words and return the length of the longest word.

Pseudocode :

```
1. Read a set of words seprated by space as str
2.  Convert  str into a list and stored into words using split function
3. Initialize length=0
4. for each i in words do
        if len(i)>length then
            Set longestword=i
            Set length=len(i)
        end if
   end for
5. Print Longest word and its length
```

Method :

Function	Description	Syntax
len()	returns the number of characters in the string.	len(object)

Source Code :

```
str=input("Enter the list of words seprated by space : ")
words=str.split()
length=0
for i in words:
    if len(i)>length:
```

```
                longestword=i
                length=len(i)
print(f"The longest word is {longestword} and its length is {length}")
```

Output:

```
Enter the list of words seprated by space : My name is Nafia
The longest word is Nafia and its length is 5
```

Result :

The program is successfully executed and the output is verified.

Experiment No :12

Date : 21/10/2024

Aim :

Write a program to prompt the user to enter two lists of integers and check

(a) Whether lists are of the same length.

(b) Whether the list sums to the same value.

(c) Whether any value occurs in both Lists.

Pseudocode :

```
1.Read two list of numbers list1 and list2.
2.//check if the lengths of the two lists are equal.
        if  length(list1)=length(list2) then
                Print the lists are same length
        else:
                Print the lists are not same length
        end If
3.Print sum of elements in both list1 and list2 using sum() method.
4.//check sum of the two lists are same.
        if sum(list1)=sum(list2) then
                Print the sum of the lists are same Else:
                Print the sum of lists are not same
         end If
5. Print common values in both list1 and list2 using set() method.
6. Set Common= set(list1)=sum(list2)
7.//check if the common list is empty.
        If  common then
                Print common value in both lists
         else:
```

```
        Print no common values in both lists
        end If
```

Method :

Function	Description	Syntax
len()	returns the number of characters in the string.	len(object)
sum()	returns a number, the sum of all items in an iterable.	sum(iterable, start)
set()	returns set of unique elements from the provided iterable.	set(iterable)

Source Code :

```
n=int(input("Enter number of integers to input for first list : "))
list1=[]
for i in range(n):
    num1=int(input("Enter Integers:"))
    list1.append(num1)
m=int(input("Enter number of integers to input for second list : "))
list2=[]
for i in range(m):
    num2=int(input("Enter Integers : "))
    list2.append(num2)
if len(list1)==len(list2):
    print("Lists are of same length")
```

```
else:
        print("Lists are of different length")
if sum(list1)==sum(list2):
        print("The lists have same sum value")
else:
        print("The lists have different sum value")
common=set(list1) & set(list2)
if common:
  print(f"Common values in both lists are:{common}")
else:
        print("There are no common value in both lists")
```

Output :

```
Enter the number of integers to input for first list : 4
Enter Integers : 1
Enter Integers : 2
Enter Integers : 3
Enter Integers : 4
Enter the number of integers to input for second list : 4
Enter Integers : 5
Enter Integers : 6
Enter Integers : 7
Enter Integers : 8
Lists are of same length
The lists have different sum value
There are no common value in both lists

Enter the number of integers to input for first list:3
Enter Integers:3
Enter Integers:2
Enter Integers:1
Enter the number of integers to input for second list:2
Enter Integers:4
Enter Integers:2
Lists are of different length
The lists have same sum value
Common value in both list: {2}

Enter the number of integers to input for first list:1
Enter Integers:2
Enter the number of integers to input for second list:3
Enter Integers:1
Enter Integers:8
Enter Integers:7
Lists are of different length
The lists have different sum value
There are no common value in both lists
```

Result :

The program is successfully executed and the output is verified.

Experiment No :13

Date : 21/10/2024

Aim :

Write a Python program to count the occurrences of each word in a line of text.

Hint: use split() function and dictionary

Pseudocode :

```
1.Read a line of text as text
2.Convert text into list using split function and store as words
3.Initialize dictionary word_count={}
4.for each i in words do
    Convert each i into lowercase
    if i in word_count then
            word_count[i]+=1
    else:
            word_count[i]=1
    end if
end for
5.Print word_count
```

Source Code :

```
text=input("Enter the text:")
words=text.split()
wordcount={}
for i in words:
    i=i.lower()
    if i in word_count:
        word_count[i]+=1
    else:
            word_count[i]=1
print(f"Word occurences are:{wordcount}")
```

Output :

```
Enter the text : My name is Nafia
Word occurrences are:{‘My’: 1 , ‘name’: 1 , ‘is’: 1 , ‘Nafia’: 1 }
```

Result :

The program is successfully executed and the output is verified.

Experiment No :14

Date : 21/10/2024

Aim :

List comprehensions:

(a) Generate positive list of numbers from a given list of integers

(b) Square of N numbers

(c) Form a list of vowels selected from a given word

(d) Form a list ordinal value of each element of a word (Hint: use ord() to get ordinal values)

Pseudocode :

```
1.  Read a list of integers as numbers.
2.  Initialize list, positive_numbers=[]
3.  for each num in list do
          If value > 0 then
            Append value to positive_numbers
           end If
    end For
4.  Print positive_numbers
5.  Set N=4
6.  square= [n**2 For num from 1 to N+1]
7.  Print square
8.  Read a string, word
9.  Convert word into lowercase as words
    10.     Initialize list, vowels=[]
    for char in word do
            if char in 'aeiou' then
```

```
                    Append char to vowels
            end If
    end For
11. Read another word
12. ordinalvalue= [ord(char) for char in word]
13. Print ordinalvalue
```

Method :

Function	Description	Syntax
ord()	returns the number representing the unicode code of a specified character.	ord(character)

Source Code :

```
numbers=[-70,15,8,3,4,-9.0,-2]
positive_numbers=[num for num in numbers if num>0]
print(f"Positive numbers:{positive_numbers}")
N=5
square=[num**2 for num in range(1,N+1)]
print("Square of first N numbers: ",square)
word="helloworld"
words=word.lower()
vowels=[char for char in word if char in 'aeiou']
print(f"Vowels in the word:{vowels}")
word="hello"
ordinalvalue=[ord(char) for char in word]
```

```
print(f"Ordinal values of each character:{ordinalvalue}")
```

Output :

```
Positive numbers: [15, 8, 3, 4]]
Squares of first N numbers: [1, 4, 9, 16, 25]
Vowels in the word:['e', 'o', 'o']
Ordinal values of each character: [104, 101, 108, 108, 111]
```

Result :

The program is successfully executed and the output is verified.

Experiment No :15

Date : 21/10/2024

Aim :

Sort dictionary in ascending and descending order.

Pseudocode :

```
1.Initialize a dictionary with key-value pairs as person1
2.Extract the values from dictionary using values() and sort in ascending order using sorted() function
3.Extract the values from dictionary using values() and sort in descending order using sorted() with reverse=True
4.Print sorted value
```

Method :

Function	Description	Syntax
sorted()	sort the list,dictionary or anything ascending by default. reverse=True will sort the list descending. Default is reverse=False	list.sorted(reverse=True| False, key=myFunc)

Source Code :

```
person1={'name1’:‘nafia’ ,’name2’:‘shahana’ ,‘name3’:’rasheed’,
‘name4’:’ashik’ }
print(f"Ascending order : {sorted(person1.values())}")
print(f"Descending order : {sorted(person1.values(),reverse=True)}")
```

Output :

```
Ascending order : [‘ashik’ ,‘nafia’, ’rasheed’, ’shahana’]
Descending order : [‘shahana’ , ‘rasheed’, ‘nafia’, ‘ashik’ ]
```

Result :

The program is successfully executed and the output is verified.

Experiment No :16

Date : 21/10/2024

Aim :

Merge two dictionaries.

Pseudocode :

```
1. Initialize  dictionary named dict1 as key-value pair
2. Initialize  dictionary named dict2 as key-value pair
3. Merge two dictionaries using update()
4. Print dict1
```

Method :

Function	Description	Syntax
update()	inserts the specified items to the dictionary.	dictionary.update (iterable)

Source Code :

```
dict1={'white’:2,’blue’:4}
dict2={'red’:3,’purple’:5}
dict1.update(dict2)
print(dict1)
```

Output :

```
{‘white’:2, ‘blue’: 4, ‘red’: 3, ‘purple’: 5}
```

Result :

The program is successfully executed and the output is verified.