Ex No: 1 Basics

1.1 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.

Aim: To 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.

Description

input() is used for taking the input from the user in the string format and float () is used for converting the possible strings into float values.

Program:

```
kilo_grams = float(input('Enter weight in Kg to Convert into pounds:'))

pounds = kilo_grams * 2.2

print(kilo_grams,' Kilograms =', pounds,' Pounds')
```

Output:

```
Enter weight in Kg to Convert into pounds:17

17.0  Kilograms = 37.400000000000006  Pounds
```

Result: Thus, the program that asks the user for a weight in kilograms and converts it to pounds is executed successfully.

Viva Questions:

1. How to convert string value into integer?.

Answer: Python uses int().

2.Is there any variation in print () function the python 2 and python3

Yes

Ex No: 1.2

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.

Aim: To 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:

```
n1=int(input("Enter 1st number:"))

n2=int(input("Enter 2nd number:"))

n3=int(input("Enter 3rd number:"))

total=n1+n2+n3

average=total/3

print("Total=",total)

print("Average=",average)
```

Output:

```
Enter 1st number:10

Enter 2nd number:20

Enter 3rd number:30

Total= 60

Average= 20.0
```

Result: Thus, the program that asks the user for three numbers and find the total and average of those numbers and printing the total and average is executed successfully.

1. How to convert string value into int with different bases?.

Answer: Python uses int(number,base).

2.Difference between / and //

Yes

Ex No: 2 Control Flow

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

Aim: To 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:

```
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,
```

Result: Thus, the program that uses a for loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89. is executed successfully.

Ex No:2.2

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

Aim: To write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.

Program:

```
name=input("Enter your name:")
times=int(input("Enter number of times to print the name:"))
for i in range(1,times+1):
   print(name)
```

Output:

```
Enter your name:aishu
Enter number of times to print the name:5
aishu
aishu
aishu
aishu
aishu
```

Result: Thus, program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times is executed successfully.

Viva Questions:

1. What is the syntax of for loop in Python?

Answer: The for loop in Python is used to iterate over a sequence (list, tuple, string) or other iterable objects. Iterating over a sequence is called traversal.

Syntax of for Loop

```
for val in sequence:
          Body of for
```

2. What is the use of range function in Python?

Answer: We can generate a sequence of numbers using range() function. range(10) will generate numbers from 0 to 9 (10 numbers).

Ex No:3 Control Flow Continued

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

```
*
**
***
****
```

Aim: To write a program using for loop to print a triangle taking the height of the triangle from the user.

Program:

```
height=int(input("Enter the height of the triangle:"))
for i in range(height+1):
   for j in range(i):
      print('*',end=" ")
   print('')
```

Output:

```
Enter the height of the triangle:4


*
```

```
* *
* * *
* * * *
```

Result: Thus,program using for loop to print a triangle taking the height of the triangle from the useris executed successfully.

Ex No:3.2

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.

Aim: To write a program to 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
num=random.randint(1,10)
guess=int(input("Enter any number from 1-10:"))
print("Random number generated=",num)
if guess==num:
    print("The random number and the guess number are same")
else:
    print("The random number and the guess number are not same")
```

Output1:

```
Enter any number from 1-10:6
Random number generated= 6
The random number and the guess number are same
```

Output2:

```
Enter any number from 1-10:6
Random number generated= 10
The random number and the guess number are not same
```

Result: Thus,program to 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 is executed successfully.

Ex No:3.3

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

Aim: To 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:

```
num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

if abs(num1 - num2) <= 0.001:

  print("Close")

else:

  print("Not close")
```

Output:

```
Enter the first number: 15

Enter the second number: 15.001

Close
```

Result: Thus,program to 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 is executed successfully.

Viva Questions:

1. What are different keywords are used for conditional statements in python?

Answer: if, else and elif

2. Write the Syntax of If statement in Python.

Answer: The syntax of if statement in Python is pretty simple.

```
if condition:

    block_of_code
```

3. Write the Syntax of If statement in Python

Answer:

Syntax of if...else

```
if test expression:
    Body of if
else:
    Body of else
```

4. Write the Syntax of If statement in Python

Answer:

The syntax of the nested *if...elif...else* construct may be –

```
if expression1:
   statement(s)
   if expression2:
      statement(s)
   elif expression3:
      statement(s)
   elif expression4:
      statement(s)
   else:
      statement(s)
else:
   statement(s)
```

Ex No: 4 Strings

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

Aim: To write a program that asks the user to enter a word and prints out whether that word contains any vowels.

Program:

```
word=input("Enter a word: ")

vowelcount=0

for i in word:

    if i in ('a,e,i,o,u,A,E,I,O,U'):

        vowelcount+=1

if vowelcount!=0:

    print("Your word contains vowel")

else:

    print("Your word has only consonants")
```

Output:

```
Enter a word: Apple

Your word contains vowel
```

Result: Thus,program to write a program that asks the user to enter a word and prints out whether that word contains any vowels is executed successfully.

Viva Questions:

1. What is a string in Python.

Answer: A string is a sequence of characters.

2. Is Python Sequences allows negative Indexing?

Answer: Python allows negative indexing for its sequences.

The index of -1 refers to the last item, -2 to the second last item and so on. We can access a range of items in a string by using the slicing operator :(colon).

3. Is it possible to remove the characters of a string?

Answer: We cannot delete or remove characters from a string. But deleting the string entirely is possible using the del keyword.

4. In Python strings are mutable or Immutable?

Answer: strings are immutable Objects.

5. How to concatenate 2 strings in Python.

Answer: The operator + is used to concatenate 2 strings.

Example: s1=”Aditya” s2 = “College” s1+s2 is “AdityaCollege”

Ex No:4.2

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

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:

```
str1=input("Enter first string:")

str2=input("Enter second string:")

str3=""

if len(str1)!=len(str2):

   print("Both strings are not having same length so cannot altered")

else:

   for i in range(0,len(str1)):

      str3=str3+str2[i]+str1[i]

print(str3)
```

Output:

```
Enter first string:abcde

Enter second string:ABCDE
```

```
AaBbCcDdEe
```

Result: Thus,program to 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 is executed successfully.

Viva Questions:

1. What is a split() method in Python?

Answer: The **split()** method returns a list of all the words in the string, using str as the separator (splits on all whitespace if left unspecified), optionally limiting the number of splits to num.

Syntax : str.split(separator, maxsplit)

2. What is join() method in Python?

Answer: The join() method is a string method and returns a string in which the elements of sequence have been joined by str separator.

Syntax: string_name.join(iterable)

3. What is len() method in Python?

Answer: The len() function returns the number of items in an object. When the object is a string, the len() function returns the number of characters in the string.

Ex No: 4.3

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.

Aim: To 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:

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

print("The number seperated with commas is:{:,}".format(n))
```

Output:

```
Enter a number:1000000

The number seperated with commas is:1,000,000
```

Result: Thus,program to 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 is executed successfully.

Ex No: 4.4

In algebraic expressions, the symbol for multiplication is often left out, as in 3x+4y or 3(x+5). Computers prefer those expressions to include the multiplication symbol, like 3*x+4*y or 3*(x+5). Write a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate

Aim: Towrite a program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate

Program:

```
AlgExpr = input("Enter algebric expression: ")

ConvertedExpr = ''

for ch in AlgExpr:

    if ch>='0' and ch<='9':

        ConvertedExpr = ConvertedExpr + ch

    elif ch=='(':

        ConvertedExpr = ConvertedExpr + '*' + ch

    elif ch>='a' and ch<='z' and ConvertedExpr[-1]!='(':

        ConvertedExpr = ConvertedExpr + '*' + ch

    else:

        ConvertedExpr = ConvertedExpr + ch

print("Converted expression is :",ConvertedExpr)
```

Output:

```
Enter algebric expression: 3x+4y

Converted expression is : 3*x+4*y
```

Result: Thus,program that asks the user for an algebraic expression and then inserts multiplication symbols where appropriate is executed successfully.

Ex No: 5. Data structure

5.1) 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.

Aim: To 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
numList=[]
for i in range(20):
   numList.append(random.randint(1,100));
print("List is :",numList)
print("Average is :",sum(numList)/20)
print("Largest element is",sorted(numList)[-1],"and smallest element is",sorted(numList)[0])
print("Second largest element is",sorted(numList)[-2],"and second smallest element is",sorted(numList)[1])
count=0
for ele in numList:
   if ele%2==0:
      count+=1
print("Total number of even elements are :",count)
```

Output:

```
List is : [52, 82, 55, 12, 70, 90, 76, 19, 77, 18, 38, 3, 87, 26, 30, 22, 71, 18, 66, 82]
```

```
Average is : 49.7

Largest element is 90 and smallest element is 3

Second largest element is 87 and second smallest element is 12

Total number of even elements are : 14
```

Result: Thus,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 is executed successfully.

Viva Questions:

1. What is the role of append() method in Python?

Answer: To add the elements at the end of the list.

Syntax : L.append(element)

2. Is the list datastructure allows duplicates or not?

Answer: Yes,allows duplicated.

Ex No: 5.2

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

Aim: To 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 a number : "))

factors=[]

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

    if num%i==0:

        factors.append(i)

print("Factors of",num,"are :",*factors,end="")
```

Output:

```
Enter a number : 8

Factors of 8 are : 1 2 4 8
```

Result: Thus,program that asks the user for an integer and creates a list that consists of the factors of that integer is executed successfully.

Viva Questions:

1. How to create an empty list?

Answer: The empty list is created by square braces[].

2. Which method is used to covert other data type into list?

Answer: list() method is used to create any type into list data type.

Ex No: 5.3

Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4.

Aim: Towrite a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4.

Program:

```
import random
l = []
def my_list():
    for j in range(0,100):
        x = random.randint(0,1)
        l.append(x)
    print (l)
    return l
def largest_row_of_zeros(l):
    c = 0
    max_count = 0
    for j in l:
        if j == 0:
            c += 1
        else:
            if c > max_count:
                max_count = c
            c = 0
    return max_count
l = my_list()
print(largest_row_of_zeros(l))
```

Output:

```
[1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0,
1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1]

5
```

Result: Thus, program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a rowis executed successfully.

Viva Questions:

1. which package is required to generate the random numbers?

Answer: import random

2. How to generate and random numbers between 0 and 100?

Answer: random.randint(0,100)

Ex No:6. Data Structure-Continued

6.1Write 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].

Aim: Towrite 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:

```
n=int(input("Enter number of elements to be insert: "))

eles=[]

print("Enter {} elements : ".format(n))

for i in range(n):

    eles.append(int(input()))

for i in range(n-1):

    j=i+1

    while j<n:

        if eles[i]==eles[j]:

            del eles[j]

            j-=1

            n-=1

        j+=1

print("After removing duplicates list is",eles)
```

Output:

```
Enter number of elements to be insert: 8

Enter 8 elements :

1

1

2

3

4

3

0

0

After removing duplicates list is [1, 2, 3, 4, 0]
```

Result: Thus, 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] is executed successfully.

Viva Questions:

1. How to remove duplicate elements from the list without writing logic?

Answer: convert the list into set data structure

2. How the elements are stored in the set data structure?

Answer: random order

Ex No: 6.2

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

Aim: To 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 a length in feet: "))

inches = feet * 12

yards = feet * 0.33333

miles = feet * 0.000189393939

millimeters = feet * 304.8

centimeters = feet * 30.48

meters = feet * 0.3048

kilometers = feet * 0.0003048

print(""" Choose 1 to convert into inches,

        choose 2 to convert into yards,
        choose 3 to convert into miles,
        choose 4 to convert into millimeters,
        choose 5 to convert into centimeters,
        choose 6 to convert into meters,
        choose 7 to convert into kilometers""")
integer = int(input("> "))
convert = [feet,inches,yards,miles,millimeters,centimeters,meters,kilometers]
print(convert[integer])
```

<u>Output:</u>

```
Enter a length in feet: 6
 Choose 1 to convert into inches,
          choose 2 to convert into yards,
          choose 3 to convert into miles,
          choose 4 to convert into millimeters,
          choose 5 to convert into centimeters,
          choose 6 to convert into meters,
          choose 7 to convert into kilometers
> 4
1828.8000000000002
```

<u>Result:</u>Thus, program to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometersis executed successfully.

Ex No: 6.3

Write a python script to perform following operations:

i) Create a matrix and print it

ii) Perform Addition of 2 matrices

iii)Perform multiplication of 2 matrices

<u>Aim:</u>To write a python script to perform following operations:

i) Create a matrix and print it

ii) Perform Addition of 2 matrices

iii)Perform multiplication of 2 matrices

<u>Program</u>: i) Create a matrix and print it

```
R = int(input("Enter the number of rows:"))
C = int(input("Enter the number of columns:"))
matrix = []
print("Enter the entries rowwise:")
for i in range(R):
    a =[]
    for j in range(C):
         a.append(int(input()))
    matrix.append(a)
for i in range(R):
    for j in range(C):
        print(matrix[i][j], end = " ")
    print()
```

<u>Output:</u>

```
Enter the number of rows:2
Enter the number of columns:2
Enter the entries rowwise:
1
1
1
1
1 1
1 1
```

<u>Program:</u>ii) Perform Addition of 2 matrices

```
R = int(input("Enter the number of rows:"))
C = int(input("Enter the number of columns:"))
A = []
print("Enter the entries rowwise:")
for i in range(R):
    a =[]
    for j in range(C):
         a.append(int(input()))
    A.append(a)
print("Matrix A:")
for i in range(R):
    for j in range(C):
        print(A[i][j], end = " ")
    print()
R1 = int(input("Enter the number of rows:"))
C1 = int(input("Enter the number of columns:"))
B = []
print("Enter the entries rowwise:")
for i in range(R1):
    b =[]
    for j in range(C1):
         b.append(int(input()))
    B.append(b)
print("Matrix B:")
for i in range(R1):
    for j in range(C1):
        print(B[i][j], end = " ")
    print()
```

```
add=[]
if R==R1 and C==C1:
    for i in range(len(A)):
        result = []
        for j in range(len(A[0])):
            result.append(A[i][j] + B[i][j])
        add.append(result)
    print("After addition new matrix is:")
    for i in range(R1):
        for j in range(C1):
            print(add[i][j],end=" ")
        print()
else:
    print("matrix addition not possible")
```

Output:

```
Enter the number of rows:2
Enter the number of columns:3
Enter the entries rowwise:
1
1
1
1
1
1
Matrix A:
1 1 1
1 1 1
Enter the number of rows:2
Enter the number of columns:2
Enter the entries rowwise:
1
1
1
1
Matrix B:
1 1
1 1
matrix addition not possible
```

Program:iii)Perform multiplication of 2 matrices

```
print("Enter order of 1st matrix:")
m,n = list(map(int,input().split()))
print("Enter Row wise values")
mat1 = []
for i in range(m) :
    print("Enter row",i,"value:")
    row = list(map(int,input().split()))
    mat1.append(row)
print("Enter order of 2nd matrix:")
p,q = list(map(int,input().split()))
print("Enter Row wise values")
mat2 = []
for j in range(p) :
    print("Enter row",j,"value:")
    row = list(map(int,input().split()))
    mat2.append(row)
print("Matrix 1:",mat1)
print("Matrix 2:",mat2)
if n==p:
    print("After Matrix Multiplication: ")
    resultant = [[ sum(ele1*ele2 for ele1,ele2 in zip(row,col)) for col in zip(*mat2) ] for row in mat1 ]
    for row in resultant:
        print(row)
else:
    print("Matrix multiplication not possible")
```

Output:

```
Enter order of 1st matrix:
2 2
Enter Row wise values
Enter row 0 value:
2 2
Enter row 1 value:
2 2
Enter order of 2nd matrix:
2 2
Enter Row wise values
Enter row 0 value:
1 1
Enter row 1 value:
1 1
Matrix 1: [[2, 2], [2, 2]]
Matrix 2: [[1, 1], [1, 1]]
After Matrix Multiplication:
```

```
[4, 4]

[4, 4]
```

Result: Thus, program to write a python script to perform following operations, Create a matrix and print it, Perform Addition of 2 matrices, Perform multiplication of 2 matrices is executed successfully.

Ex No: 7 Functions

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

Aim: To 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):
   sd=0
   while num!=0:
      sd=sd+num%10
      num=num//10
   return sd
num=int(input("Enter number: "))
print("Sum of digits of {} is {}".format(num,sum_digits(num)))
```

Output:

```
Enter number: 12056

Sum of digits of 12056 is 14
```

Result: Thus, program to write a function called sum_digits that is given an integer num and returns the sum of the digits of num is executed successfully.

Viva Questions:

1. 1.What is the keyword used to declare a Function?

   Answer: The def keyword is used to declare Function

2. What is parameterized function?

Answer: Declaring a function with at least one parameter.

Ex No: 7.2

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.

Aim: To 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:
       if len(s1)==len(s2):
          for i in range(len(s1)):
              if s1[i]!=s2[i]:
                  return (i+1)


s1=input("Enter string 1: ")
s2=input("Enter string 2: ")
x=first_diff(s1,s2)
if(x== -1):
    print("strings are identical")
else:
    print("first difference occurs at location :",x)
```

Output:

```
Enter string 1: apple
Enter string 2: appve
first difference occurs at location : 4
```

Result: Thus, program to 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 is executed successfully.

1.What is the keyword used to declare a Function?

Answer: The def keyword is used to declare Function

2.What is lambda in Python?

Answer: It is a single expression anonymous function often used as inline function.

3. Write the syntax for creating a function in python?

Answer: def function_name(arguments.....):

Body of the function....

Ex No: 7.3

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

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

Program:

```
def number_of_factors(n):
    result = []
    for i in range(1, n + 1):
        if n % i == 0:
            result.append(i)
    return len(result)
n=int(input("enter a number"))
print("No.of factors of ",n,":",number_of_factors(n))
```

Output:

```
enter a number15

No.of factors of  15 : 4
```

Result: Thus, program to write a function called number_of_factors that takes an integer and returns how many factors the number has is executed successfully.

Ex No: 7.4

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

Aim: To 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):
    f=0
    i=1
    while i<len(l):
        if l[i]<l[i-1]:
            f=1
        i+=1
    if not f:
        return True
    else:
        return False
l=[]
n=int(input("enter list size:"))
for i in range(0,n):
    el=int(input())
    l.append(el)
print(is_sorted(l))
```

Output:

```
enter list size:5
1
2
3
5
4
False
```

Result: Thus, program to write a function called is_sorted that is given a list and returns True if the list is sorted and False otherwise is executed successfully.

Ex No: 8 Functions-Continued

8.1 Write a function called root that is given a number x and an integer n and returns x1/n . In the function definition, set the default value of n to 2.

Aim: To write a function called root that is given a number x and an integer n and returns x1/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(int(input("enter integer")),int(input("enter n value:"))))
print(root(int(input("enter integer"))))
```

Output:

```
enter integer8
enter n value:2
2.8284271247461903
enter integer4
2.0
```

Result: Thus, program to write a function called root that is given a number x and an integer n and returns x1/n In the function definition, set the default value of n to 2 is executed successfully.

Ex No: 8.2

Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.

Aim: To write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.

Program:

```
def primes(n=100):
    p=[]
    i=1
    for k in range(1,n+1):
        c = 0
        for j in range(1,i+1):
            a=i%j
            if a==0:
                c+=1
        if c==2:
            p.append(i)
        else:
            k=k-1
        i=i+1
    return p
n= int(input("Enter the number:"))
print("prime numbers of first ",n," primes " ,primes(n))
print("primes for default values of n",primes())
```

Output:

```
Enter the number:50
```

```
prime numbers of first  50  primes  [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

primes for default values of n [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Result: Thus, program to write a function called primes that is given a number n and returns a list of the first n primes is executed successfully.

1. What is docstring in Python?

Answer: A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes

2. What is lambda in Python?

Answer: It is a single expression anonymous function often used as inline function.

Ex No: 8.3

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

Aim: To 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:

```
def merges(l1,l2):
   return(sorted(l1+l2))
test_list1 = [1, 5, 6, 9, 11]
```

```
test_list2 = [3, 4, 7, 8, 10]
print ("The combined sorted list using sort method is : ",merges(test_list1,test_list2))


def merge(ls1,ls2):
    size_1=len(ls1)
    size_2=len(ls2)
    res = []
    i, j = 0, 0
    while i < size_1 and j < size_2:
        if ls1[i] < ls2[j]:
            res.append(ls1[i])
            i += 1
        else:
            res.append(ls2[j])
            j += 1
    res = res + ls1[i:] + ls2[j:]
    return res
test_list1=[1, 5, 6, 9, 11]
test_list2=[3, 4, 7, 8, 10]
print("The combined sorted list without using sort method is : ",merge(test_list1,test_list2))
```

Output:

```
The combined sorted list using sort method is :  [1, 3, 4, 5, 6, 7, 8, 9, 10, 11]
The combined sorted list without using sort method is :  [1, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

Result: Thus, program to write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sorted list,Do this using the sort method, Do this without using the sort method is executed successfully

Ex No: 8.4

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.

Aim: To 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:

```
Enter a word: car

ca cr ac ar rc ra
```

Result: Thus, 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 is executed successfully

Ex No: 9 Files

9.1 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.

Aim: To 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 file name: "),"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
abcd@gmail.com;pqrs@gmail.com;google@gmail.com;youtube@gmail.com
```

Result: Thus, program to 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 is executed successfully

1. What is docstring in Python?

Answer: A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes

2. What is lambda in Python?

Answer: It is a single expression anonymous function often used as inline function.

Ex No: 9.2

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.

Aim: To 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()
print("contents in temps.txt:\n")
print(lines)
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()
ftemps=open("ftemps.txt","r")
contents=ftemps.read()
print("Successfully converted and copied the results to file ftemps.txt\n")
print("contents in ftemps.txt:\n")
print(contents)
ftemps.close()
```

Output:

```
contents in temps.txt:

['45.7\n', '44.5\n', '44.8\n', '43.3\n', '43.2\n', '44.1']
Successfully converted and copied the results to file ftemps.txt

contents in ftemps.txt:

114.26
112.1
112.64
109.94
109.76
111.38
```

Result: Thus, program to 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 is executed successfully

Viva Questions:

1.What is File?

Answer: A file is some information or data which stays in the computer storage devices. ... Python gives you easy ways to manipulate these files. Generally we divide files in two categories, text file and binary file. Text files are simple text where as the binary files contain binary data which is only readable by computer.

2.What is a file handle in Python?

Answer:Python has a built-in open() function to open a file. This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.

3.The readlines() method returns ____________

Answer: A list of lines

4.Which command is used to open a file "c:\temp.txt" in append-mode

Answer: outfile = open("c:/temp.txt", "a")

5.How split() function works?

Answer: Split a string into a list where each word is a list item

Ex No: 9.3

Write a program to count frequency of characters in a given file.

Aim: To write a program to count frequency of characters in a given file.

Program:

```
file = open("text.txt","r")
a=[]
b={}
for i in file:
    for j in range(0,len(i)):
        a.append(i[j])
for i in a:
    if i in b:
        b[i]+=1
    else:
        b[i]=1
print(b)
```

Output:

```
{'T': 2, 'h': 6, 'i': 9, 's': 8, ' ': 21, 'a': 7, 'm': 2, 'p': 3, 'l': 5, 'e': 12, 't': 8, 'x': 1, 'f': 2, '.': 3, '\n': 2, 'U': 1,
'n': 5, 'g': 2, 'u': 3, 'w': 2, 'r': 7, 'P': 1, 'y': 2, 'o': 2, 'c': 2, 'd': 1, 'b': 1}
```

Result: Thus, program to count frequency of characters in a given file is executed successfully

Ex No: 10 OOP

10.1 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 a 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.

Aim: To 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 a 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, amount, price):
        self.name = name
        self.amount = amount
        self.price = price
    def get_price(self, number_to_be_bought):
        discount = 0
        if number_to_be_bought < 10:
            pass
        elif 10 <= number_to_be_bought < 99:
            discount = 10
        else:
            discount = 20
        price = (100 - discount) / 100 * self.price
        return price * number_to_be_bought
    def make_purchase(self, quantity):
        self.amount -= quantity
name, amount, price = 'shoes', 200, 33
shoes = Product(name, amount, price)
q1 = 4
print(f'cost for {q1} {shoes.name} = {shoes.get_price(q1)}')
shoes.make_purchase(q1)
print(f'remaining stock: {shoes.amount}\n')
q2 = 12
print(f'cost for {q2} {shoes.name} = {shoes.get_price(q2)}')
shoes.make_purchase(q2)
print(f'remaining stock: {shoes.amount}\n')
q3 = 112
print(f'cost for {q3} {shoes.name} = {shoes.get_price(q3)}')
shoes.make_purchase(q3)
print(f'remaining stock: {shoes.amount}\n')
```

Output:

```
cost for 4 shoes = 132.0
remaining stock: 196

cost for 12 shoes = 356.4
remaining stock: 184

cost for 112 shoes = 2956.8
remaining stock: 72
```

Result: Thus, program to 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 a 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 is executed successfully

1.What is Object Oreinted Programming?

Answer: OOP stands for Object-Oriented Programming. Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions. OOP provides a clear structure for the programs.

2.What is an object?

Answer: Any thing that exist in the real world physically is called an object.

3. What is a class?

Answer: A group of similar objects are called class. A class is a blue print for creating objects.

Ex No: 10.2

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.

Aim: To 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, seconds):
       self.seconds = seconds
   def convert_to_minutes(self):
       mins = self.seconds//60
       secs = self.seconds - (mins*60)
       return ("%d:%d" %(mins,secs))
   def convert_to_hours(self):
       secs = self.seconds
       hours = secs//3600
       secs = secs - (hours*3600)
       mins = secs//60
       secs = secs - (mins*60)
       return ("%d:%d:%d" %(hours,mins,secs))
time = Time(int(input()))
print(time.convert_to_minutes())
time = Time(int(input()))
print(time.convert_to_hours())
```

Output:

```
240
4:0
4520
1:15:20
```

Result: Thus, program to 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 is executed successfully

4. **What is data abstraction?**

Answer: Hiding of unwanted data and show the essential one is called data abstraction.

5. **What is inheritance?**

Answer: Inheritance is a process of creating a new class from existing class and acquire all the properties of its parent.

6. **What is syntax for creating a class in python?**

Answer:

Syntax:

```
class <class-name>:
        data members
        functions....

Object=classname()
```

Ex No: 10.3

Write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, c = Converter(9,'inches'). The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the Converter object created above, the user could call c.feet() and should get 0.75 as the result.

Aim: To write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, c = Converter(9,'inches'). The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the Converter object created above, the user could call c.feet() and should get 0.75 as the result.

Program:

```
class Converter:
    def __init__(self,length,unit):
        self.length=length
        self.unit=unit
    def feet(self):
        if(self.unit=='feet'):
            return self.length
        elif(self.unit=='inches'):
            return self.length/12
        elif(self.unit=='yards'):
            return self.length/0.333
        elif(self.unit=='miles'):
            return self.length/0.000189
        elif(self.unit=='millimeters'):
            return self.length/304.8
        elif(self.unit=='centimeters'):
            return self.length/30.48
        elif(self.unit=='meters'):
            return self.length/0.305
        elif(self.unit=='kilometers'):
            return self.length/0.000305
    def inches(self):
        if(self.unit=='feet'):
            return self.length*12
        elif(self.unit=='inches'):
            return self.length
        elif(self.unit=='yards'):
            return self.length*36
        elif(self.unit=='miles'):
            return self.length*63360
        elif(self.unit=='millimeters'):
            return self.length*0.0393701
        elif(self.unit=='centimeters'):
            return self.length*0.393701
        elif(self.unit=='meters'):
            return self.length*39.3701
        elif(self.unit=='kilometers'):
            return self.length*39370.1
    def yards(self):
        if(self.unit=='feet'):
            return self.length*0.333333
        elif(self.unit=='inches'):
            return self.length*0.0277778
        elif(self.unit=='yards'):
            return self.length
        elif(self.unit=='miles'):
            return self.length*1760
        elif(self.unit=='millimeters'):
            return self.length*0.00109361
        elif(self.unit=='centimeters'):
            return self.length*0.0109361
        elif(self.unit=='meters'):
            return self.length*1.09361
        elif(self.unit=='kilometers'):
            return self.length*1093.61

    def miles(self):
        if(self.unit=='feet'):
            return self.length*0.000189394
        elif(self.unit=='inches'):
            return self.length*63360
        elif(self.unit=='yards'):
            return self.length*0.027777728
        elif(self.unit=='miles'):
            return self.length
        elif(self.unit=='millimeters'):
            return self.length/1609344
        elif(self.unit=='centimeters'):
            return self.length/160934.4
        elif(self.unit=='meters'):
            return self.length/1609.344
        elif(self.unit=='kilometers'):
            return self.length/1.609
    def kilometers(self):
        if(self.unit=='feet'):
            return self.length/3280.84
        elif(self.unit=='inches'):
            return self.length/39370.1
        elif(self.unit=='yards'):
            return self.length/1093.61
        elif(self.unit=='miles'):
            return self.length/0.621371
        elif(self.unit=='millimeters'):
            return self.length/1000000
        elif(self.unit=='centimeters'):
            return self.length/100000
        elif(self.unit=='meters'):
            return self.length/1000
        elif(self.unit=='kilometers'):
            return self.length
    def meters(self):
        if(self.unit=='feet'):
            return self.length/3.28084
        elif(self.unit=='inches'):
            return self.length/39.3701
        elif(self.unit=='yards'):
            return self.length/1.09361
        elif(self.unit=='miles'):
            return self.length/0.000621371
        elif(self.unit=='millimeters'):
            return self.length/1000
        elif(self.unit=='centimeters'):
            return self.length/100
        elif(self.unit=='meters'):
            return self.length
        elif(self.unit=='kilometers'):
            return self.length/0.001
    def centimeters(self):
        if(self.unit=='feet'):
            return self.length/0.0328084
        elif(self.unit=='inches'):
            return self.length/0.393701
        elif(self.unit=='yards'):
            return self.length/0.0109361
        elif(self.unit=='miles'):
            return self.length*160934
        elif(self.unit=='millimeters'):
            return self.length/10
        elif(self.unit=='centimeters'):
            return self.length
        elif(self.unit=='meters'):
            return self.length*100
        elif(self.unit=='kilometers'):
            return self.length*100000
    def millimeters(self):
        if(self.unit=='feet'):
            return self.length*304.8
        elif(self.unit=='inches'):
            return self.length/0.0393701
        elif(self.unit=='yards'):
            return self.length/0.00109361
        elif(self.unit=='miles'):
            return self.length*1609340
        elif(self.unit=='millimeters'):
            return self.length
        elif(self.unit=='centimeters'):
            return self.length*10
        elif(self.unit=='meters'):
            return self.length*100
        elif(self.unit=='kilometers'):
            return self.length*1000000

len=int(input("Enter length: "))
type=input("Enter unit type: inches,feet,yards,miles,millimeters,centimeters,meters,kilometers---> ")
c=Converter(len,type)
print("Length in Feet: ",round(c.feet(),3))
print("Length in Inches: ",round(c.inches(),3))
print("Length in Yards: ",round(c.yards(),3))
print("Length in Miles: ",round(c.miles(),3))
print("Length in Kilometers: ",round(c.kilometers(),3))
print("Length in Meters: ",round(c.meters(),3))
print("Length in Centimeters: ",round(c.centimeters(),3))
print("Length in Millimeters: ",round(c.millimeters(),3))
```

Output:

```
Enter length: 9
Enter unit type: inches,feet,yards,miles,millimeters,centimeters,meters,kilometers---> inches
Length in Feet:  0.75
Length in Inches:  9
Length in Yards:  0.25
Length in Miles:  570240
Length in Kilometers:  0.0
Length in Meters:  0.229
Length in Centimeters:  22.86
```

Length in Millimeters: 228.6

Result: Thus, program to write a class called Converter. The user will pass a length and a unit when declaring an object from the class—for example, c = Converter(9,'inches'). The possible units are inches, feet, yards, miles, kilometers, meters, centimeters, and millimeters. For each of these units there should be a method that returns the length converted into those units. For example, using the Converter object created above, the user could call c.feet() and should get 0.75 as the result is executed successfully

Ex No: 11 OOP Continued

11.1 Write a Python class to implement pow(x, n)

Aim: To write a Python class to implement pow(x, n)

Program:

```
#11.1
class Power:
  def pow(self, x, n):
      if x==0 or x==1 or n==1:
         return x

      if x==-1:
         if n%2 ==0:
            return 1
         else:
            return -1
      if n==0:
         return 1
      if n<0:
         return 1/self.pow(x,-n)
      val = self.pow(x,n//2)
      if n%2 ==0:
         return val*val
```

```
        return val*val*x
p=Power()
print(p.pow(int(input("enter number")), int(input("enter power"))))
print(p.pow(3, 5))
print(p.pow(100, 0))
```

Output:

```
enter number8
enter power2
64
243
1
```

Result: Thus, program to write a Python class to implement pow(x, n) is executed successfully

Ex No: 11.2

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

Aim: To write a Python class to reverse a string word by word.

Program:

```
class Reverse:
    def reverse_words(self, s):
        return ' '.join(reversed(s.split()))
print(Reverse().reverse_words(input("enter the string: ")))
```

Output:

```
enter the string: welcome to python programming
programming python to welcome
```

Result: Thus, program to write a Python class to reverse a string word by word is executed successfully.

Ex No: 12 GUI & Exception Handling

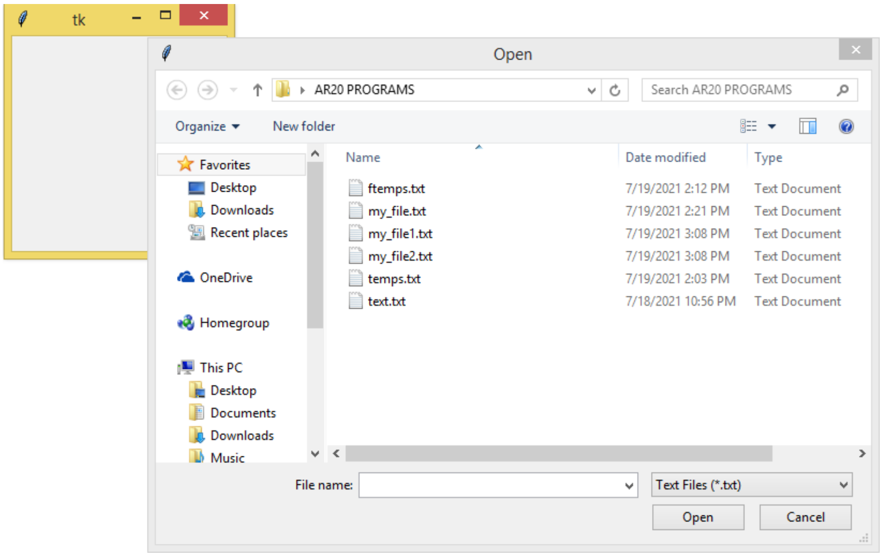
12.1 Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox.

Aim: To write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox.

Program:

```
from tkinter import filedialog

from tkinter import Tk

from tkinter import *

root = Tk()

root.fileName = filedialog.askopenfilename(filetypes=(("Text Files",".txt"),("All Files","*.*")))

text1 = open(root.fileName).read()

T = Text(root, height=25, width=80)

T.pack()

T.insert(END,text1) #END (or “end”) corresponds to the position just after the last character in the buffer.

root.mainloop()
```

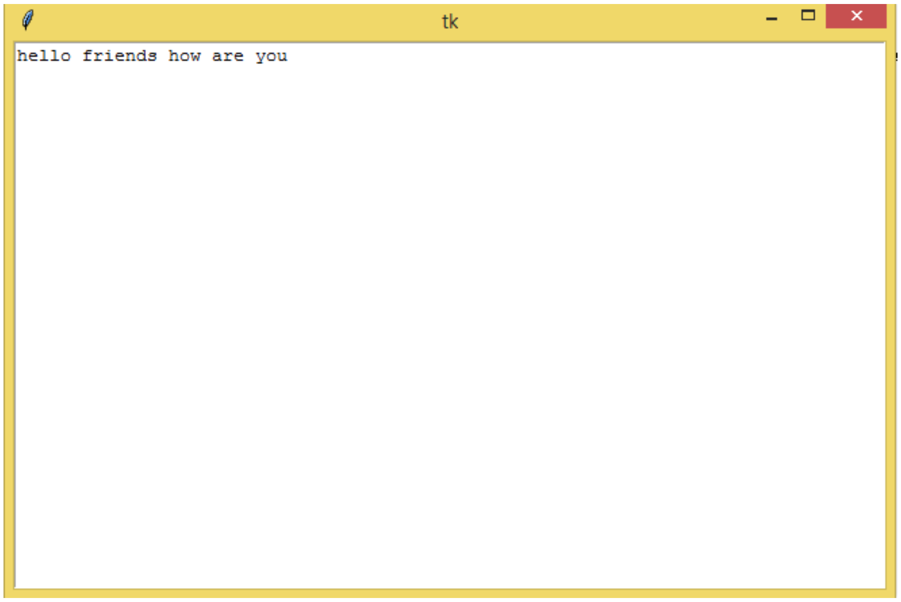
Output:



Name	Date modified	Type
ftemps.txt	7/19/2021 2:12 PM	Text Document
my_file.txt	7/19/2021 2:21 PM	Text Document
my_file1.txt	7/19/2021 3:08 PM	Text Document
my_file2.txt	7/19/2021 3:08 PM	Text Document
temps.txt	7/19/2021 2:03 PM	Text Document
text.txt	7/18/2021 10:56 PM	Text Document

File name:

Text Files (*.txt)



```
hello friends how are you
```

Result: Thus, program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox is executed successfully.

Viva Questions:

1.What is Exception?

Answer: An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions

2.What is difference between Exception and Error?

Answer: Errors mostly occur at runtime that's they belong to an unchecked type. Exceptions are the problems which can occur at runtime and compile time.

Ex No:12.2

Write a program to demonstrate Try/except/else.

Aim: To 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 not possible when b=0")
else:
    print("a//b value :",c)
```

Output 1:

```
Enter 'a' value: 10
Enter 'b' value: 0
Division not possible when b=0
```

Output 2:

```
Enter 'a' value: 10
Enter 'b' value: 2
a//b value : 5
```

Result: Thus, program to demonstrate try/except/else is executed successfully.

3. What are try,except,finally keywords?

Answer: Try a block of code, and decide what to to if it raises an error:

```
try:
                                x                              > 3
except:
  print("Something went wrong")
```

The except keyword is used in try...except blocks. It defines a block of code to run if the try block raises an error.You can define different blocks for different error types, and blocks to execute if nothing went wrong, see examples below.

The finally keyword is used in try...except blocks. It defines a block of code to run when the try...except...else block is final. The finally block will be executed no matter if the try block raises an error or not.This can be useful to close objects and clean up resources.

4.How to define User defined exception?

Answer:. When we are developing a large Python program, it is a good practice to place all the user-defined exceptions that our program raises in a separate file.

Syntax:

```
# define Python user-defined exceptions
class Error(Exception):
    """Base class for other exceptions"""
    pass



class ValueTooSmallError(Error):
    """Raised when the input value is too small"""
    pass
```

```
class ValueTooLargeError(Error):
    """Raised when the input value is too large"""
    pass
```

Ex No:12.3

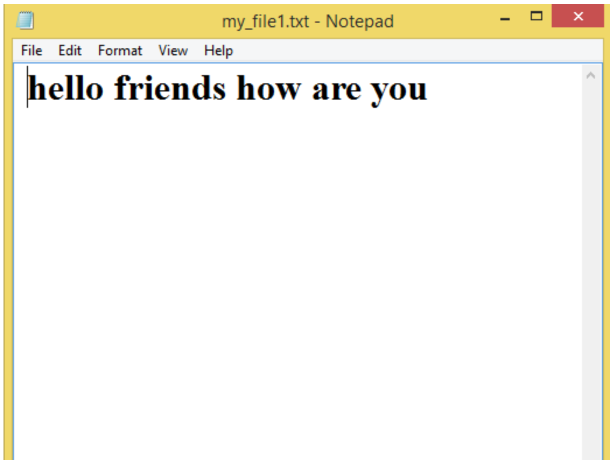
Write a program to demonstrate try/finally and with/as.

Aim: To write a program to demonstrate try/finally and with/as.

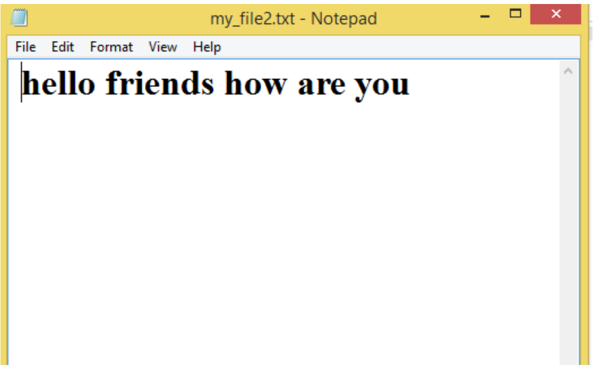
Program:

```
file=open("my_file1.txt","w")
try:
    file.write("hello friends how are you")
finally:
    file.close()
with open("my_file2.txt","w") as file:
    file.write("hello friends how are you")
```

OUTPUT:



```
hello friends how are you
```



hello friends how are you

Result: Thus, program to demonstrate try/finally and with/as is executed successfully.

Aim: Guess the Number

The Goal: This project uses the random module in Python. The program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to input information.) If the user's guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear. You'll need functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers.

```
import random

number=random.randrange(0,100)

guessCheck="wrong"

print("Welcome to Number Guess")

l="Yes"

while l=="Yes":

        response=int(input("Please input a number between 0 and 100:"))

        try:

                val=int(response)

        except ValueError:

                print("This is not a valid integer. Please try again")

                continue

        val=int (response)

        if val<number:

                print("This is lower than actual number. Please try again.")
                l=raw_input("Do you want to continue!!!Yes or No ")

        elif val>number:
                print("This is higher than actual number. Please try again.")
                l=raw_input("Do you want to continue!!!Yes or No ")
        else:
                print("This is the correct number")
                l="No"
if l=="No":
    print "Number in the game is ",number
print("Thank you for playing Number Guess. See you again")

'''
TestCase1:
Welcome to Number Guess
Please input a number between 0 and 100:345
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No No
Number in the game is  90

Thank you for playing Number Guess. See you again
TestCase2:
Welcome to Number Guess
Please input a number between 0 and 100:56
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:43
This is higher than actual number. Please try again.
```

```
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:34
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:23
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:20
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:19
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:18
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:15
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:10
This is higher than actual number. Please try again.
Do you want to continue!!!Yes or No Yes
Please input a number between 0 and 100:5
This is the correct number
Number in the game is  5
Thank you for playing Number Guess. See you again
'''
```

Hangman

Aim:

The Goal: Despite the name, the actual “hangman” part isn’t necessary. The main goal here is to create a sort of “guess the word” game. The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you’ll need a way to grab a word to use for guessing. (This can be grabbed from a pre-made list. No need to get too fancy.) You will also need functions to check if the user has actually inputted a single letter, to check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses.

Description:

This is a Python script of the classic game “Hangman”. The word to guess is represented by a row of dashes. If the player guess a letter which exists in the word, the script writes it in all its correct positions. The player has 10 turns to guess the word. You can easily customize the game by changing the variables.

Program:

```
import time

name = input("What is your name? ")

print ("Hello, " + name, "Time to play hangman!")

print ("")

time.sleep(1)

print ("Start guessing...")

time.sleep(0.5)

word = "secret"

guesses = ''

turns = 10

while turns > 0:

    failed = 0

    for char in word:
        if char in guesses:
            print (char, )
        else:
            print ("_",)
            failed += 1
    if failed == 0:
        print ("You won!")
        break
    print()
    guess = input("guess a character:")
    guesses += guess
    if guess not in word:
        turns -= 1
        print ("Wrong")
        print ("You have", + turns, 'more guesses' )
        if turns == 0:
            print ("You Lose")
```

Output:

```
What is your name? vinay
Hello, vinay Time to play hangman!

Start guessing...
_
_
_
_
_
_
```

```
guess a character:s
s
–
–
–
–
–

guess a character:e
s
e
–
–
e
–

guess a character:c
s
e
c
–
e
–

guess a character:r
s
e
```

```
c
r
e
–

guess a character:e
s
e
c
r
e
–

guess a character:t
s
e
c
r
e
t
You won!
```

Result: Thus, the program to guess the letters of a word is executed successfully.

Viva Questions:

1) What is a time.sleep() in Python?

Answer: The method **sleep()** suspends execution for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time.The actual suspension time may be less than that requested because any caught signal will terminate the sleep() following execution of that signal's catching routine.

Syntax:

Following is the syntax for **sleep()** method −

```
time.sleep(t)
```

Exp.No.15 Write a program to find the greatest number that can be formed by using given set of numbers

Aim: Write a program to find the greatest number that can be formed by using given set of numbers

Description:

.

Naive Approach: The naive approach is to sort the given array of digits in **descending order** and then form the number using the digits in array keeping the order of digits in the number same as that of the sorted array.

Given an array of integers arr[] representing digits of a number. The task is to write a program to generate the largest number possible using these digits.

Note: The digits in the array are in between 0 and 9. That is, 0<arr[i]<9.

Examples:

Input : arr[] = {4, 7, 9, 2, 3}

Output : Largest number: 97432

Input : arr[] = {8, 6, 0, 4, 6, 4, 2, 7}

Output : Largest number: 87664420

Program:

```
def findMaxNum(arr,n) :


    # sort the given array in
    # descending order
    arr.sort(reverse = True)


    # initialize num with starting
    # element of an arr
    num = arr[0]


    # generate the number
    for i in range(1,n) :
        num = num * 10 + arr[i]


    return num
```

```
# Driver code
if __name__ == "__main__" :

    arr = [1,2,3,4,5,0]

    n = len(arr)

    print(findMaxNum(arr,n))
```

Output:

```
543210
```

Result: Thus, the program to find the greatest number that can be formed by using given set of numbers is executed successfully.

Viva Questions:

1. What is sort method?

Answer: The **sort**() **method sorts** the elements of a given list in a specific ascending or descending order. The syntax of the **sort**() **method** is: ... **sort**(key=..., reverse=...) Alternatively, you can also use **Python's** built-in **sorted**() **function** for the same purpose.

Exp.No.16 Write a program to find sum of digits of a number till you get single digit sum.

Aim: Write a program to find sum of digits of a number till you get single digit sum.

Example:

Input : 142 (Hint: 1+4+2=7)

Ouput : 7

Input : 4683 (Hint: 4+6+8+3=21 =>2+1=3)

Ouput : 3

Description:

.

To implement this ,we used While Loop. A **while** loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

Syntax:

The syntax of a **while** loop in Python programming language is −

```
while expression:
      statement(s)
```

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any non-zero value. The loop iterates while the condition is true.

Program:

```
n=int(input("enter x"))
b=n
s=n
while s>9:
   n=s
   s=0
   while n>0:
      r=n%10
      s=s+r
      n//=10
print('sum of digits of number till you get single digit{} is {}'.format(b,s))
```

Output:

```
enter x142
sum of digits of number till you get single digit142 is 7

enter x4683
sum of digits of number till you get single digit4683 is 3
```

Result: Thus, the program to to find sum of digits of a number till you get single digit sum executed successfully.

Viva Questions:

1. for i in range(-3), how many times this loop will run ?

Answer:1

2. for loop in python are work on

Answer: Range and Iteration

3. To break the infinite loop , which keyword we use ?

Answer:break

4**.** What we put at the last of the loop ?

Answer:colon

5. How many times it will print the statement ?, for i in range(100): print(i)

Answer:100

Exp.No.17 Write a program to count how many times each word present in a file.

AIM: To write a program to count how many times each word present in a file.

Description:

Many times it is required to count the occurrence of each word in a text file. To achieve so, we make use of a dictionary object that stores the word as the key and its count as the corresponding value. We iterate through each word in the file and add it to the dictionary with count as 1. If the word is already present in the dictionary we increment its count by 1.

Program:

```
# Open the file in read mode

text = open("file.txt", "r")


# Create an empty dictionary

d = dict()
```

```
# Loop through each line of the file
for line in text:
    # Remove the leading spaces and newline character
    line = line.strip()

    # Convert the characters in line to
    # lowercase to avoid case mismatch
    line = line.lower()

    # Split the line into words
    words = line.split(" ")

    # Iterate over each word in line
    for word in words:
        # Check if the word is already in dictionary
        if word in d:
            # Increment count of word by 1
            d[word] = d[word] + 1
        else:
            # Add the word to dictionary with count 1
            d[word] = 1

# Print the contents of dictionary
for key in list(d.keys()):
    print(key, ":", d[key])
```

Output:

```
python : 5
is : 3
an : 1
interpreted, : 1
high-level : 1
and : 7
general-purpose : 1
programming : 2
language. : 2
created : 2
by : 1
guido : 1
van : 1
rossum : 1
first : 1
released : 2
in : 3
1991, : 1
python's : 1
design : 1
philosophy : 1
emphasizes : 1
code : 2
readability : 1
with : 2
its : 3
```

notable : 1

Result: Thus, the program to count how many times each word present in a file is executed successfully.

Viva Questions:

1.What is dictionary?

Answer: A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

2.What is split() function?

Answer: The **split**() method splits a string into a list. You can specify the separator, default separator is any whitespace.

3. What will be the output of the following Python code snippet?

```
1.   d = {"john":40, "peter":45}
2.   d["john"]
```

Answer:40

4.Dictionary keys must be immutable

Answer:False