

WEST GODAVARI INSTITUTE OF SCIENCE AND ENGINEERING
Prakasaraopalem, Avapadu, Tadepalligudem, W.G.Dist

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PYTHON PROGRAMMING
LABORATORY MANUAL

IB.Tech, II Semester
Common to CSE , AIDS (**R20**)
(2020– 2023 Batch)



Prepared by:

K. Chandra Sekhar
Assistant Professor
Department of ECE

Approved by:

P.Sheela
Associate Professor
Head of the Department CSE

Academic Year: 2022-2023



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

I Year - II Semester		L	T	P	C
		0	0	3	1.5
PYTHON PROGRAMMING LAB					

Course Objectives:

The Objectives of Python Programming are

- *To learn about Python programming language syntax, semantics, and the runtime environment*
- *To be familiarized with universal computer programming concepts like data types, containers*
- *To be familiarized with general computer programming concepts like conditional execution, loops & functions*
- *To be familiarized with general coding techniques and object-oriented programming*

Course Outcomes:

- Develop essential programming skills in computer programming concepts like data types, containers
- Apply the basics of programming in the Python language
- Solve coding tasks related conditional execution, loops
- Solve coding tasks related to the fundamental notions and techniques used in object-oriented programming

List of Experiments:

- 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.
- 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.
- 3) Write a program that uses a *for* loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.
- 4) 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.
- 5) Use a *for* loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.

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

- 6) 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.
- 7) 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.
- 8) Write a program that asks the user to enter a word and prints out whether that word contains any vowels.
- 9) 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



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

- 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*.
- 10) 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.
 - 11) 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.
 - 12) 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.
 - 13) Write a program that asks the user for an integer and creates a list that consists of the factors of that integer.
 - 14) 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.
 - 15) Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list $[1,1,2,3,4,3,0,0]$ would become $[1,2,3,4,0]$.
 - 16) 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.
 - 17) Write a function called *sum_digits* that is given an integer num and returns the sum of the digits of num.
 - 18) 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.
 - 19) Write a function called *number_of_factors* that takes an integer and returns how many factors the number has.
 - 20) Write a function called *is_sorted* that is given a list and returns True if the list is sorted and False otherwise.
 - 21) Write a function called *root* that is given a number x and an integer n and returns $x^{1/n}$. In the function definition, set the default value of n to 2.
 - 22) 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.
 - 23) 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.
 - 24) 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.
 - 25) 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.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

- 26) 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*.
- 27) 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.
- 28) 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.
- 29) 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.
- 30) Write a Python class to implement *pow(x, n)*.
- 31) Write a Python class to reverse a string word by word.
- 32) 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.
- 33) Write a program to demonstrate *Try/except/else*.
- 34) Write a program to demonstrate *try/finally* and *with/as*.



WEST GODAVARI INSTITUTE OF SCIENCE AND ENGINEERING
Prakasaraopalem, Avapadu, Tadepalligudem, W.G.Dist

PYTHON LABORATORY

VISION & MISSION

Vision	To produce innovative, ethical and socially responsible graduates in the field of Electronics and Communication Engineering who can address the global and Environmental challenges through excellence in technical education and Research.
Mission	To provide creative learning environment to the students by offering state-of-art infrastructure, core instruction and highly qualified and motivated faculty. To adapt the learners to use new technologies in Electronics and Communication Engineering field by engaging in cutting-edge research through collaborations with industry and academia. To imbibe self-learning attitude, entrepreneur skills, and professional ethics and leadership qualities among the graduates to make them globally competent. To extend the benefits of the technology to solve the environmental and Societal challenges.

PROGRAM EDUCATIONAL OBJECTIVES

PEO No.	PEO Description
PEO 1	Develop a solid foundation in Mathematics, Science and Technology to solve Computer Science and Engineering problems.
PEO 2	Identify, analyze and apply core engineering concepts to develop novel products and solutions for real life problems.
PEO 3	Pursue higher studies, research & development and other creative efforts in science & technology and keep abreast of latest technological developments.
PEO 4	Inculcate professional and ethical attitude, effective communication skills and team spirit and leadership qualities.
PEO 5	Contribute to the needs of the society in solving technical problems using Computer Science and Engineering principles, tools and practices.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PYTHON LABORATORY

I B. Tech. II Semester CSE (R20) PROGRAM OUTCOMES

POs	PO STATEMENT
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization for the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, research literature, and analyses complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources and modern engineering and IT tools, including prediction and modeling to complex engineering activities, with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
PSO1	Professional Skills: An Ability to Understand the concepts of basic Python program and to apply them to various areas like data analysis data science, Etc.
PSO2	Problem Solving skills: An ability to solve the problems in programs using the python tools and their built in functions.
PSO3	Successful Career and Entrepreneurship: Wisdom of social and environmental awareness along with ethical responsibility to have a successful career and to sustain passion and zeal for real-world applications using optimal resources as an Entrepreneur.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GENERAL INSTRUCTIONS:

1. The experiments have been designed to be performed within the 3-hour laboratory time.
2. To successfully complete the experiment in one lab turn, come prepared to the laboratory.
3. Read the experiment in advance.
4. To Install python lab software.
5. Open the Idle shell and go to file menu to open new script.
6. In the new script to write python programs. And save it go run.
7. Record stepwise observe the outputs for each program.
8. Get the observation signed by the instructor.
9. Always take safety precautions with using computers.

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.

Code:

```
kilograms=float(input("Enter Weight in Kilograms: "))
pounds=2.2*kilograms
print("Weight in Pounds: ",pounds)
```

OUTPUT 1:

```
Enter Weight in Kilograms: 600
Weight in Pounds: 1320.
```

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.

Code:

```
n1=int(input("Enter First Number: "))
n2=int(input("Enter Second Number: "))
n3=int(input("Enter Third Number: "))
total=n1+n2+n3
average=total/3
print("Total= ",total)
print("Average= ",average)
```

OUTPUT 2:

```
Enter First Number: 50
Enter Second Number: 65
Enter Third Number: 78
Total= 193
Average= 64.33333333333333
```

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

Code:

```
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
```

4.) 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.

Code:

```
name=input("Enter your Name: ")
n=int(input("Enter how many times you want to print: "))
for i in range(n):
    print(name)
```

OUTPUT 4:

```
Enter your Name: ANIL
Enter how many times you want to print: 9
ANIL
ANIL
ANIL
ANIL
ANIL
ANIL
ANIL
ANIL
ANIL
```

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

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

Code:

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

OUTPUT:

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

6. 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.

Code:

```
import random
n=random.randint(1,10)
usernumner=int(input("Enter a number between 1 to 10(inclusive): "))
if(n==usernumner):
    print("Your Guess is Right")
else:
    print("Your Guess is Wrong")
print("Random Number: ",n)
```

OUTPUT :

```
Enter a number between 1 to 10(inclusive): 5
Your Guess is Wrong
Random Number: 10
```

7.) 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.

Code:

```
from decimal import *
num1 = Decimal(input("Enter number 1 : "))
num2 = Decimal(input("Enter number 2 : "))
diff= abs(num1-num2)
if(diff<=0.001):
    print("Close")
else:
    print("Not Close")
```

OUTPUT 7:

```
Enter number 1 : 18.098
Enter number 2 : 30.990
Not Close
```

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

Code:

```
word=input("Enter a Word: ")
vowel='aeiouAEIOU'
flag=0
for i in vowel:
    if i in word:
        flag=1
        break
if flag==1:
    print("Word Contain Vowel")
else:
    print("Word not Contain Vowel")
```

OUTPUT 8:

```
Enter a Word: CRY
Word not Contain Vowel
```

9.) 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.

Code:

```
s1=input("Enter First String: ")
s2=input("Enter Second String: ")
if(len(s1)!=len(s2)):
    print("String are with same length")
    result=""
    for i in range(len(s1)):
        result=result+(s2[i]+s1[i])
    print(result)
else:
    print("String are with different length")
```

OUTPUT 9:

```
Enter First String: abcdef
Enter Second String: ABCDEF
String are with same length
AaBbCcDdEeFf
```

10. 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.

Code:

```
number=int(input("Enter a Long Number: "))
print("{:,}".format(number))
```

OUTPUT :

```
Enter a Long Number: 18500005432300
18,500,005,432,30
```

11.) 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.

CODE:

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

OUTPUT:

```
enter algebraic expression:3x+4y
3*x+y*y
enter algebraic expression:3(x+5)
3*(x+5)
```


12. 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

CODE:

```
import random
l=[]
for i in range(20):
    l.append(random.randint(1,100))
print("List: ",l)
print("Average: ",
      round(sum(l)/len(l),2))
print("Largest Value in List: ",max(l))
print("Smallest Value in List: ",min(l))
l1=sorted(l)
print( " Second Largest Value in List: ",l1[-2])
print("Smallest Value in List: ",l1[1])
count=0
for i in l:
    if i%2==0:
        count+=1
print("Number of Even Numbers in the list: ",count)
```

OUTPUT:

```
List: [69, 11, 20, 55, 7, 33, 94, 38, 42, 82, 26, 32, 15, 78, 81, 78, 57, 22, 50, 46]
Average: 46.8
Largest Value in List: 94
Smallest Value in List: 7
Second Largest Value in List: 82
Smallest Value in List: 11
Number of Even Numbers in the list: 12
```

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

CODE:

```
n=int(input("Enter a number: "))
l=[]
for i in range(1,n+1):
    if(n%i==0):
        l.append(i)
print(l)
```

OUTPUT:

```
Enter a number: 4
[1, 2, 4]
```

14. 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.

CODE:

```
import random
x=[]
for i in range(100):
    x.append(random.randint(0,1))
maxzero=0
count=0
for i in range(len(x)):
    if(x[i]==0):
        count=count+1

    if(i==len(x)-1):
        if(count>maxzero):
            maxzero=count
    if(x[i]==1):
        if(count>maxzero):
            maxzero=count
        count=0
print("Longest run of Zeros in a row is",maxzero)
```

OUTPUT:

```
Longest run of Zeros in a row is 7
```

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

CODE:

```
l=list(map(int,input("Enter the elements into list with duplication: ").split(',')))
s=[]
for i in l:
    if i not in s:
        s.append(i)
print(s)
```

OUTPUT:

```
Enter the elements into list with duplication: 1,1,2,56,78,56,1
[1, 2, 56, 78]
```

16. 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.

CODE:

```
feet=int(input("Enter feet: "))
opt=int(input("enter choice 1:inches 2:yards 3:miles 4:millimeters 5:centimeters 6:meters
7:kilometers --->"))
l=[round(feet*12,3),round(feet*0.333,3),round(feet*0.000189,3),round(feet*304.8,3),\
round(feet*30.48,3),round(feet*0.305,3),round(feet*0.000305,3)]
print(l[opt-1])
```

OUTPUT:

```
enter feet:6
enter choice1:inches 2:yards 3:miles 4:millimetres 5:centimeters 6:meters 7:kilometres ---> 7
0.02
```

OR

PROGRAM2:

CODE:

```
feet=int(input("Enter feet:"))
opt=int(input("enter choice 1:inches 2:yards 3:miles 4:millimeters 5:centimeters 6:meters
7:kilometers --->"))
if(opt==1):
    print(round(feet*12,3))
elif(opt==2):
    print(round(feet*0.333,3))
elif(opt==3):
    print(round(feet*0.00189,3))
elif(opt==4):
    print(round(feet*304.8,3))
elif(opt==5):
    print(round(feet*30.48,3))
elif(opt==6):
    print(round(feet*0.305,3))
elif(opt==7):
    print(round(feet*0.000305,3))
```

OUTPUT:

```
enter feet:6
enter choice 1:inches 2:yards 3:miles 4:millimetres 5:centimeters 6:meters 7:kilometres --> 7
0.02
```

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

CODE:

```
def sum_digits(num):
    sum=0
    while(num>0):
        sum=sum+num%10
        num=num//10
    return sum
x=int(input("Enter a number: "))
s=sum_digits(x)
print("Sum of digits: ",s)
```

OUTPUT:

```
Enter a number: 153
Sum of digits: 9
```

18. 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.

CODE:

```
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: apvle
first difference occurs at location : 3
```

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

CODE:

```
def number_of_factors(n):
    fact_count=0
    for i in range(1,n+1):
        if(n%i==0):
            fact_count+=1
    return fact_count
n=int(input("Enter an integer: "))
x=number_of_factors(n)
print("factors count is",x)
```

OUTPUT:

```
Enter an integer: 14
factors count is 4
```


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

CODE:

```
def is_sorted(l):
    x=l[:]
    x.sort()
    if l==x:
        return True
    else
        return False
l=list(input("Enter list items : ").split())
print(is_sorted(l))
```

OUTPUT:

```
Enter list items : 34,56,78,99
True
```

21. Write a function called `root` that is given a number `x` and an integer `n` and returns $x^{1/n}$. In the function definition, set the default value of `n` to 2.

CODE:

```
def root(x,n=2):
    return (x**(1/n))
x=int(input("Enter 'x' value: "))
n=int(input("Enter 'n' value: "))
ans1=root(x,n)
ans2=root(x)
print("Root value with n value: ",ans1)
print("Root Value with out n value (Default 2): ",ans2)
```

OUTPUT:

```
Enter 'x' value: 16
Enter 'n' value: 4
Root value with n value: 2.0
Root Value with out n value (Default 2): 4.0
```

22. 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.

CODE:

```
def printprime(n=100):
    l=[]
    x=2
    while(len(l)<n):
        for i in range(2,int(x**0.5)+1):
            if(x%i==0):
                break
            else:
                l.append(x)
                x=x+1
    return(l)
n=int(input("Enter no of primes wanted: "))
s=printprime(n)
print("List of first",n,"primes:",s)
j=printprime()
print("List of first 100 primes:",j)
```

OUT PUT:

Enter no of primes wanted: 10

List of first 10 primes: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

List of first 100 primes: [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, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541]

23. 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.

a) Do this using the sort method

CODE:

```
def merge (l1,l2):
    l=l1+l2
    l.sort()
    return l
l1=list(map(int,input("Enter the sorted list 1:").split()))
l2=list(map(int,input("Enter the sorted list 2:").split()))
s=merge(l1,l2)
print("After merge the list is:",s)
```

OUT PUT:

Enter the sorted list 1:1 4 6 8 9

Enter the sorted list 2:2 6 8 9 90

After merge the list is: [1, 2, 4, 6, 6, 8, 8, 9, 9, 90]

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

CODE:

```
def merge_lists(L1,L2):
    # when one of them is an empty list,returns the other list

    if not L1:
        return L2
    elif not L2:
        return L1
    result=[]
    i=0
    j=0
    for k in range(len(L1)+len(L2)):
        if L1[i]<=L2[j]:
            result.append(L1[i])
            if i<len(L1)-1:
                i+=1
            else:
                result+=L2[j:] #when the last element in L1 is reached,
                break #append the rest of L2 to result.
        else:
            result.append(L2[j])
            if j<len(L2)-1:
                j+=1
            else:
                result+=L1[i:] #When the last element in L2 is reached,
                break #append the rest of L1 to result.
    return result
l1=list(map(int,input("Enter the sorted list 1:").split()))
l2=list(map(int,input("Enter the sorted list 2:").split()))
s=merge_lists(l1,l2)
print("After merge the list is :",s)
```

OUT PUT:

Enter the sorted list 1:2 3 4 78

Enter the sorted list 2:1 89 23 90

After merge the list is : [1, 2, 3, 4, 78, 89, 23, 90]

24. 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.

CODE:

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

OUT PUT:

Enter a word: CSE
CS CE SC SE EC ES

25. 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.

CODE:

```
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=";")
```

OUT PUT:

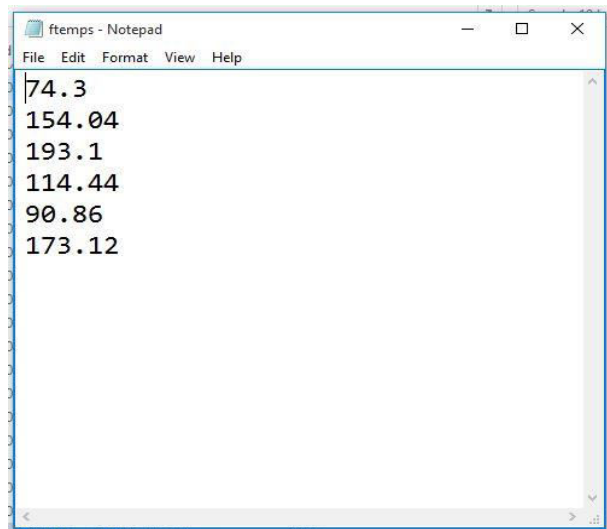
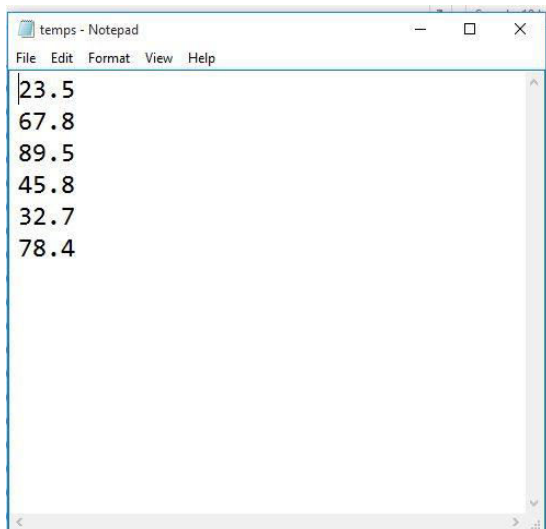
Enter file name: CSE.txt
ramana@gmail.com;raju@gmail.com;krishne@gmail.com;kiran@gmail.com;ganesh@gmail.com

26. 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.

CODE:

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

OUT PUT:



27. 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.

CODE:

```
class product:
```

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

    def get_price(self, number_to_be_bought):
        discount = 0
        if number_to_be_bought<10:
            print("Regular price is charged for your order")
            print("NO Discount\n")
            cost=self.price*number_to_be_bought
            print('Final costs = ',cost)
            print(" A 10% discount is applied for orders of between 10 and 99 items")
            print(" A 20% discount is applied for orders of 100 or more items")
        elif 10<=number_to_be_bought<99:
            Actual_cost=self.price*number_to_be_bought
            print('Actual cost is =',Actual_cost)
            discount=10
            p=(100-discount)/100*self.price
            cost=p*number_to_be_bought
            print('Final costs(Discount cost10%)is = ',cost)
            print("you save=",Actual_cost-cost,"Rupees")
        else:
```

```
Actual_cost=self.price*number_to_be_bought
print('Actual_cost is=',Actual_cost)
discount = 20
p=(100-discount)/100*self.price
cost=p*number_to_be_bought
print('Final costs(Discount cost20%) = ',cost,)
print("you save=",Actual_cost-cost,"Rupes")
```

```
name=input("Enter the Name of the product:\n")
total_items= int(input("Total Number of items:\n')),
price=int(input('Digit price of each items:\n'))
p= product(name,total_items,price)
number_to_be_bought=int(input("Enter Number of items you want u buy:"))
p.get_price(number_to_be_bought)
```

OUT PUT:

```
Enter the Name of the product:
pen
Total Number of items:
200
Digit price of each item:
5
Enter Number of items you want u buy:7
Regular price is charged for your order
NO Discount

Final costs = 35
A 10% discount is applied for orders of between 10 and 99 items
A 20% discount is applied for orders of 100 or more item
```

28. 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.

CODE:

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

OUT PUT:

```
Enter seconds: 230
Time in minutes:seconds format ---> 3:50
Time in hours:minutes:seconds format ---> 0:3:50
```

29. 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.**

CODE:

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*0.0277778
    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))
```

OUT PUT:

```
Enter length: 6
Enter unit type: inches,feet,yards,miles,millimeters,centimeters,meters,kilometers---> inches
Length in Feet: 0.5
Length in Inches: 6
Length in Yards: 0.167
Length in Miles: 380160
Length in Kilometers: 0.0
Length in Meters: 0.152
Length in Centimeters: 15.24
Length in Millimeters: 152.4
```

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

CODE:

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

OUT PUT:

```
Enter 'x' value : 2
Enter 'n' value : 4
pow( 2 , 4 ) = 16
```

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

CODE:

```
class reverse:
    def rev_sentence(self,sentence):
        words = sentence.split(' ')
        reverse_sentence = ' '.join(reversed(words))
        print(reverse_sentence)
c=reverse()
c.rev_sentence(input("Enter the string: "))
```

OUT PUT:

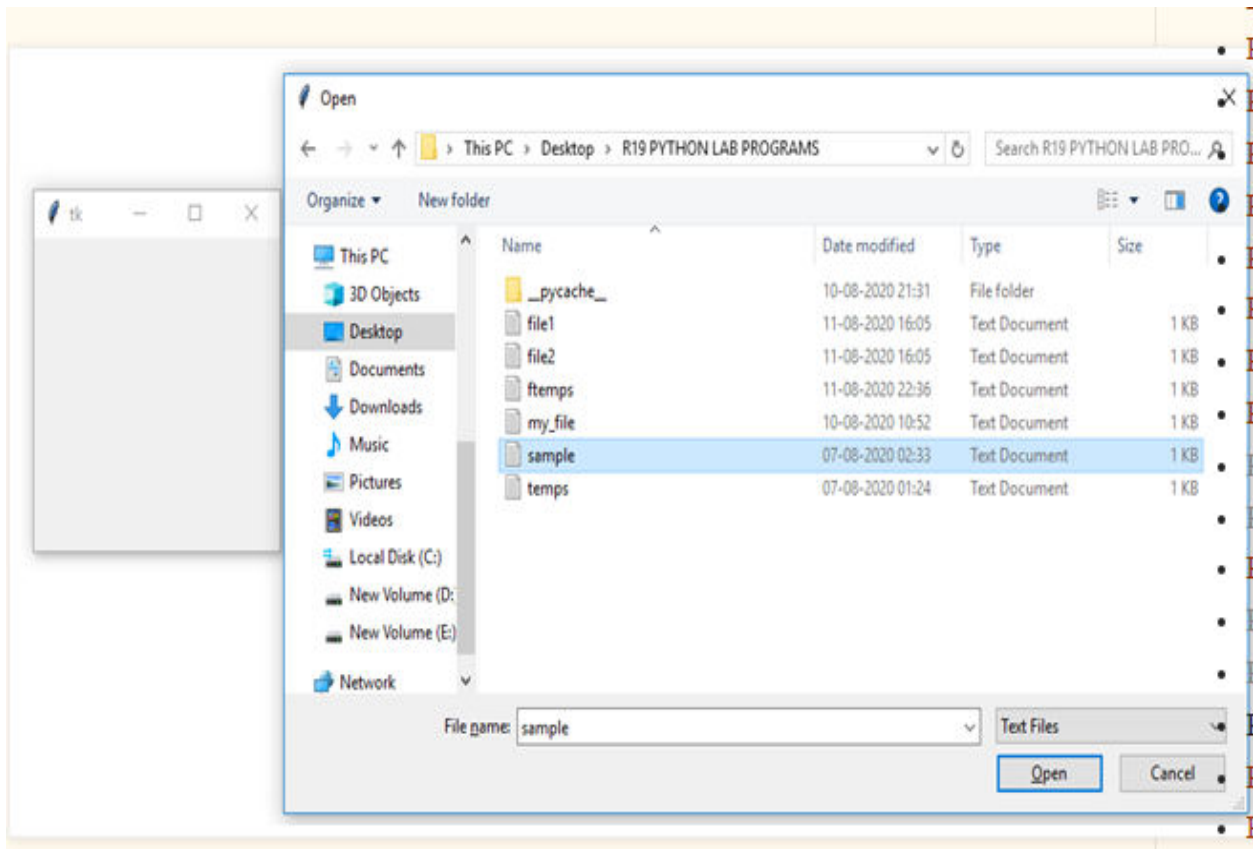
```
Enter the string: HAI FRIENDS HOW ARE YOU
YOU ARE HOW FRIENDS HAI
```

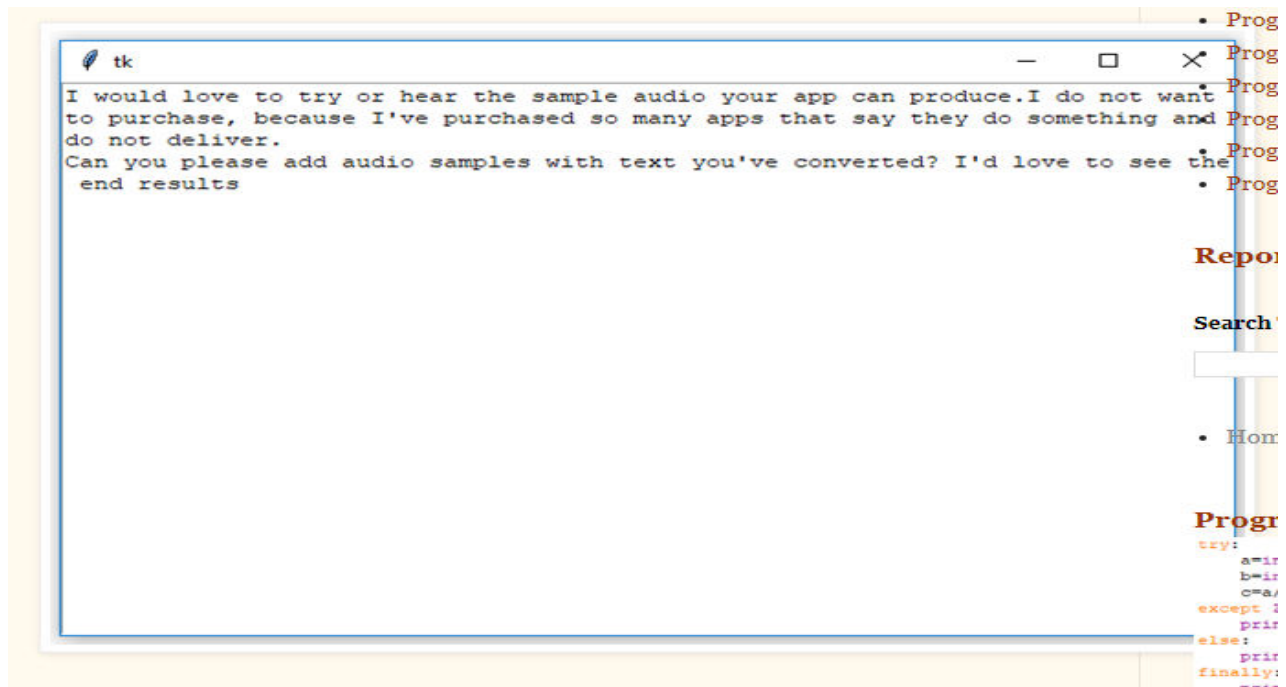
32. Write a program that opens a file dialog that allows you to select a text file. The program then displays the contents of the file in a textbox.

CODE:

```
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()
```

OUT PUT:





33. Write a program to demonstrate Try/except/else.

CODE:

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

OUT PUT:

```
Enter 'a' value: 10
Enter 'b' value: 0
Division can't possible (b=0)
```

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

CODE:

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

OUT PUT:

```
Enter 'a' value: 10
Enter 'b' value: 0
Division can't possible (b=0)
End of the program
```

(OR)

```
Enter 'a' value: 10
Enter 'b' value: 2
a//b Value: 5
End of the program
```

Or

```
file=open('file.txt', 'w')
try:
file.write('hello friends how are you')
finally:
file.close()
with open('file2.txt', 'w') as file:
file.write('hello friends how are you')
```

Output:

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

