TABLE OF CONTENTS

LAB RULES

Responsibilities of Users

Users are expected to follow some fairly obvious rules of conduct:

Always:

- Enter the lab on time and leave at proper time.
- Wait for the previous class to leave before the next class enters.
- Keep the bag outside in the respective racks.
- Utilize lab hours in the corresponding.
- Turn off the machine before leaving the lab unless a member of lab staff has specifically told you not to do so.
- Leave the labs at least as nice as you found them.
- If you notice a problem with a piece of equipment (e.g. a computer doesn't respond) or the room in general (e.g. cooling, heating, lighting) please report it to lab staff immediately. Do not attempt to fix the problem yourself.

Never:

- Don't abuse the equipment.
- Do not adjust the heat or air conditioners. If you feel the temperature is not properly set, inform lab staff; we will attempt to maintain a balance that is healthy for people and machines.
- Do not attempt to reboot a computer. Report problems to lab staff.
- Do not remove or modify any software or file without permission.
- Do not remove printers and machines from the network without being explicitly told to do so by lab staff.

- Don't monopolize equipment. If you're going to be away from your machine for more than 10 or 15 minutes, log out before leaving. This is both for the security of your account, and to ensure that others are able to use the lab resources while you are not.
- Don’t use internet, internet chat of any kind in your regular lab schedule.
- Do not download or upload of MP3, JPG or MPEG files.
- No games are allowed in the lab sessions.
- No hardware including USB drives can be connected or disconnected in the labs without prior permission of the lab in-charge.
- No food or drink is allowed in the lab or near any of the equipment. Aside from the fact that it leaves a mess and attracts pests, spilling anything on a keyboard or other piece of computer equipment could cause permanent, irreparable, and costly damage. (and in fact *has*) If you need to eat or drink, take a break and do so in the canteen.
- Don’t bring any external material in the lab, except your lab record, copy and books.
- Don’t bring the mobile phones in the lab. If necessary then keep them in silence mode.
- Please be considerate of those around you, especially in terms of noise level. While labs are a natural place for conversations of all types, kindly keep the volume turned down.

If you are having problems or questions, please go to either the faculty, lab in-charge or the lab supporting staff. They will help you. We need your full support and cooperation for smooth functioning of the lab.

INSTRUCTIONS

Before entering in the lab

All the students are supposed to prepare the theory regarding the next experiment.
Students are supposed to bring the practical file and the lab copy.
Previous programs should be written in the practical file.
All the students must follow the instructions, failing which he/she may not be allowed in the lab.

While working in the lab

Adhere to experimental schedule as instructed by the lab in-charge.
Get the previously executed program signed by the instructor.
Get the output of the current program checked by the instructor in the lab copy.
Each student should work on his/her assigned computer at each turn of the lab.
Take responsibility of valuable accessories.
Concentrate on the assigned practical and do not play games
If anyone caught red handed carrying any equipment of the lab, then he/she will have to face serious consequences.

POORNIMA INSTITUTE OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
Python Lab (6CS4-23)

Zero Lab

Prepared by: Shikha Gautam

1) Name of the lab with code: Python Lab (6CS4-23)

2) Self-Introduction:

- **a) Name** : Shikha Gautam
- **b) Qualification** : M.Tech. (CS)
- **c) Designation** : Assistant Professor
- **d) Subject Taught** : DSA,SE, OS, DS, CC, OOPs, ASP.Net, DBMS, FOC, Prg in C, ERP, E-Commerce, SPM,
- **e) Lab taken** : DSA Lab, Programming in C Lab, OOPS Lab
- **f) E-mail ID** : shikha.gautam@poornima,org
- **g) Introduction of Students:**

An interactive session will be held with the students wherein they will be asked to introduce themselves, covering the following points:-

- **a)** Academics Merit/ Weak
- **b)** Co-curricular Activity
- **c)** Day Scholar/ Hosteller
- **d)** Medium Hindi/ English
- **e)** Family Background Urban/ Rural
- **f)** Learning Style seeing/ hearing/ doing

4). Introduction to Lab:-

a) **Relation with other labs:**

Students have machine learning and DIP lab in current semester to they required basic knowledge of Python for implementing their programs, in Next semester they have to develop their project so for that purpose they again require learning of python because in current scenario generally projects development teams using python language

b) **Lab schedule per week:**

Two hours per batch per week

5) University Examination System:-

Sr. No.	Name of the Exam	Max. Marks	% of passing marks	Syllabus coverage (in %)	Conducted by
1	I Mid Term Exam	40	40	50%	PGC
2	II Mid Term Exam	40	40	50%	**PGC**
3	University (End) Term Exam	40	40	100%	University

Place: PIET, Jaipur
Date : April 25, 2021

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Syllabus

III Year-VI Semester: B.Tech. Computer Science and Engineering

6CS4-23: Python Lab

Credit: 1.5 **Max. Marks: 75(IA:45, ETE:30)**
0L+0T+3P **End Term Exam: 2 Hours**

SN	List of Experiments
1	Write a program to demonstrate basic data type in python.
2	Write a program to compute distance between two points taking input from the user Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
3	Write a Program for checking whether the given number is an even number or not. Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . , 1/10
4	Write a Program to demonstrate list and tuple in python. Write a program using a for loop that loops over a sequence. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
5	Find the sum of all the primes below two million. By considering the terms in the Fibonacci sequence whose values do not exceed four million, WAP to find the sum of the even-valued terms.
6	Write a program to count the numbers of characters in the string and store them in a dictionary data structure Write a program to use split and join methods in the string and trace a birthday of a person with a dictionary data structure
7	Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file? Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?
8	Write a program to print each line of a file in reverse order. Write a program to compute the number of characters, words and lines in a file.
9	Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on. Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.
10	Write a program to implement Merge sort. Write a program to implement Selection sort, Insertion sort.

Office of Dean Academic Affairs
Rajasthan Technical University, Kota

Lab Plan

6CS4-23 PYHTON LAB

S.No.	List of Experiments
1	**Topic to be covered : Installations, Indentations, Comments, Variables, Data Types, Casting, User inputs** **Exercise Based on above Topics:** 1. Introduction of Python and installation of anaconda Jupyter by students. 2. Write a program to demonstrate basic data type in python. 3. Write a program to compute distance between two points taking input from the user 4. Write a program add.py that takes 2 numbers as input from user and prints its sum.
2	**Topics to be covered : Operators, If-else, Loops(For, While Loop)** **Exercise based on above Topics:** 1. Write a Python program to print maximum out of 3 numbers. 2. Write a Python program to compute the compound interest & simple interest 3. Write a Python program to check given number is even or not. 4. Using a for loop, write a program that prints out the decimal equivalents of ½, 1/3, 1/4… 1/10 5. Write a program using a for loop that loops over a sequence. 6. Write a program using while loop that asks the user for a number, and prints a countdown from that number to zero.
3	**Topics to be covered : List, Tuples, Set, Dictionaries** **Exercise based on above Topics:** 1. Write a program to demonstrate list in python 2. Write a program to demonstrate tuples in python 3. Write a program to demonstrate Set in python 4. Write a program to demonstrate Dictionaries in python 5. Write a program to check number is Armstrong or not 6. Write a program to check number is palindrome or not 7. Write a program to check number is prime or not
4	**Topics to be covered : Strings, Array, File Handling** **Exercise based on above Topics:** 1. Write a program to count the numbers of characters in the string and store them in a dictionary data structure 2. Write a program to use split and join methods in the string and trace a birthday of a person with a dictionary data structure 3. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file? 4. Write a program to print each line of a file in reverse order. 5. Write a program to compute the number of characters, words and lines in a file.

	6. Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on. 7. Write a program to implement Insertion sort. 8. Write a program to implement Selection sort
5	**Topics to be covered : Functions, Classes, Objects** **Exercise based on above Topics:** 1. Write a Python function to find the Max of three numbers. 2. Write a Python program to reverse a string. 3. Write a Python function to calculate the factorial of a number (a non-negative integer).The function accepts the number as an argument. 4. Write a Python class to reverse a string word by word. 5. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.
6	**Topics to be covered : Scope, Modules, Inheritance** **Exercise based on above Topics:** Demonstration through examples
After I Mid Term Experiment Number 7 to 12 to be execute	
7	**Topics to be covered : Introduction of Flask, Environment, Application, Routing** **Exercise based on above Topics:** 1.Write a program to print " hello World"
8	**Topics to be covered : Flask Routing, URL Building, HTTP Methods** **Exercise based on above Topics:** Demonstration through examples
9	**Topics to be covered : Templates, static Files, Request objects** **Exercise based on above Topics:** Demonstration through examples
10	**Topics to be covered : Sending Form Data to Templates, Cookies and Session** **Exercise based on above Topics:** Demonstration through examples
11	**Topics to be covered : Redirect & Errors, Message Flashing, File Upload** **Exercise based on above Topics:** Demonstration through examples
12	**Topics to be covered : Database Connectivity(SQLite)** **Exercise based on above Topics:** Application of Students Information form (Database Connectivity ,SQL Query : Create , Insert , Select)
Mid Term second would be application based	

MARKS SCHEME

RTU Marks Scheme

Maximum Marks Allocation		
Internal	End-Term	Total
45	30	75

Marks Division

Mid Term – I & II		
Performance	Viva	Total
30	10	40
Attendance & Performance		
Performance	Attendance	Total
30	10	40
End-Term Practical		
Performance	Viva	Total
30	10	40

Internal Assessment System

Total Marks – 10

Attendance	Discipline	Performance	Record	Viva	Total
2	2	2	2	2	10

LAB PLAN

Total number of experiment 12

Total number of turns required 12

Number of turns required for

Experiment Number	Turns	Scheduled Day
Exp. 1	1	Day 1
Exp. 2	1	Day 2
Exp. 3	1	Day 3
Exp. 4	1	Day 4
Exp. 5	1	Day 5
Exp. 6	1	Day 6
Exp. 7	1	Day 7
Exp. 8	1	Day 8
Exp. 9	1	Day 9
Exp. 10	1	Day 10
Exp. 11	1	Day 11
Exp. 12	1	Day 12

Distribution of lab hours

Attendance	05 minutes
Explanation of the concept	30 minutes
Explanation of experiment	15 minutes
Performance of experiment	45 minutes
Viva / Quiz / Queries	25 minutes
Total	**120 minutes**

Software required

Anaconda Jupyter Notebook/Google Colab on Windows Platform

Introduction to the subject

Difference between Programming and Scripting Language:

Definition

A programming language is essentially a formal language that combines a set of instructions that can be fed into the computer to generate a specific output. A scripting language is a programming language that supports scripts which are programs written exclusively for a special runtime environment to automate the execution of a specific action/function.

Interpretation

Programming languages are compiled into a more compact design that does not require to be interpreted by another language or application. Scripting languages are written in one language and interpreted within another program, for instance, JavaScript has to be incorporated within HTML which will then be interpreted by the Internet browser. Thus, programming languages run independently of a parent program, but scripting languages run inside another program.

Design

Programming languages are designed to facilitate a full-fledged code and software development whereas scripting languages are specifically designed to make coding faster and much simpler.

Development

writing a full-fledged code with programming languages usually take a longer time to develop as more lines need to be written while coding with a scripting language requires less time as smaller chunks need to be written.

Categories

Programming languages are divided into five subcategories: First generation, Second generation, Third generation, Fourth generation, and Fifth generation. Scripting languages have only two subcategories: Server-side scripting languages and client-side scripting languages.

Conversion and hosting

Since programming languages use a compiler, it is a one-shot conversion. Scripting languages, on the other hand, demand line by line conversion. Programming languages are self-executable; they do not require a host. Scripting languages require a host.

Speed

Compiled programs run generally run faster than interpreted programs since compilers read and analyze the code at once and report errors (if any) collectively. An interpreter, however, reads and analyzes a code line by line and every time it detects an error, it stops to address them one by one.

Languages

C, C++, C#, Java, Basic, COBOL, and Pascal, are some examples of programming languages. JavaScript, Perl, PHP, Python, Ruby, Rexx, Ruby, GameMonkey, etc., are some of the most widely used such languages.

.

Lab Objective

Course Outcome Required : .

Procedure

Step 1. Click on Start icon on desktop
Step 2. Open Anaconda navigator
Step 3. Launch jupyter notebook
Step 4. Open new python file
Step 5. This editor is used to write programs
Step 6. Save the program

Experiments

Experiment No.: 1

Topic to be covered : Installations, Indentations, Comments, Variables, Data Types, Casting, User inputs

Exercise Based on above Topics:

1. Introduction of Python and installation of anaconda Jupyter by students.
2. Write a program to demonstrate basic data type in python.
3. Write a program to compute distance between two points taking input from the user
4. Write a program add.py that takes 2 numbers as input from user and prints its sum.

1) Introduction of Python and installation of anaconda Jupyter by students.

Prerequisite: Python

While Jupyter runs code in many programming languages, Python is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook.

We recommend using the Anaconda distribution to install Python and Jupyter.

Installing Jupyter using Anaconda

For new users, we highly recommend installing Anaconda. Anaconda conveniently installs Python, the Jupyter Notebook

Use the following installation steps:

Download Anaconda. We recommend downloading Anaconda's latest Python 3 version (currently Python 3.5).

Install the version of Anaconda which you downloaded, following the instructions on the download page.

Congratulations, you have installed Jupyter Notebook.

2) Write a program to demonstrate basic data type in python.

```
print(type(1))
print(type(2.2))
print(type('ABCD'))
print(type([1,2,3]))
print(type({1:'a'}))
print(type((1,2,3)))
print(type({1,2,3}))

Output:
<class 'int'>
<class 'float'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'tuple'>
<class 'set'>
```

3)Write a program to compute distance between two points taking input from the user.

```
#formula : sqrt((x2 - x1)^2 + (y2 - y1)^2).
x1= int(input("enter X1 value"))
x2= int(input("enter X2 value"))
y1= int(input("enter y1 value"))
y2= int(input("enter y2 value"))
result=(((x2-x1)**2 + (y2-y1)**2)**0.5)
print("the distance between two points",(x1,x2),"and",(y1,y2),"=",result)

Output:
enter X1 value5
enter X2 value6
enter y1 value4
enter y2 value3
the distance between two points (5, 6) and (4, 3) = 1.4142135623730951
```

4)Write a program add.py that takes 2 numbers as input from user and prints its sum.

```
num1=int(input("Enter the first Number"))
num2=int(input("Enter the Second Number"))
sum=num1+num2
print("The Sum of Two numbers is", sum)

Output:
Enter the first Number5
Enter the Second Number4
The Sum of Two numbers is 9
```

Experiment No.: 2

Topics to be covered : Operators, If-else, Loops(For, While Loop)

Exercise based on above Topics:

1. Write a Python program to print maximum out of 3 numbers.
2. Write a Python program to compute the compound interest & simple interest
3. Write a Python program to check given number is even or not.
4. Using a for loop, write a program that prints out the decimal equivalents of ½, 1/3, 1/4… 1/10
5. Write a program using a for loop that loops over a sequence.
6. Write a program using while loop that asks the user for a number, and prints a countdown from that number to zero.

1. Write a Python program to print maximum out of 3 numbers.

```
num1=int(input("enter the first number"))
num2=int(input("enter the Second number"))
num3=int(input("enter the third number"))
if (num1>num2) and (num1>num3):
 print("First number is maximum",num1)
elif (num2>num1) and (num2>num3):
 print("Second number is maximum",num2)
else:
 print("Third number is maximum",num3)
```

```
Output:
  enter the first number6
  enter the Second number7
  enter the third number8
  Third number is maximum 8
```

2. Write a Python program to compute the compound interest & simple interest.

```
principle=float(input(" enter the principle amount"))
rate=float(input(" enetr the rate of interest"))
time=float(input("Enter the time period"))
amount=principle*((1+rate/100)**time)
print(amount)
```

```
Output:
  enter the principle amount5000
  enetr the rate of interest5
  Enter the time period2
   5512.5
```

3. Write a Python program to check given number is even or not.

```
number=int(input("enter the number to be check"))
if (number%2==0):
 print("number is even")
else:
 print("number is not even")

Output:
enter the number to be check5
number is not even
```

4. Using a for loop, write a program that prints out the decimal equivalents of ½, 1/3, 1/4… 1/10

```
for i in range(2,11):
 print(1/i)
Output:
0.5
0.3333333333333333
0.25
0.2
0.16666666666666666
0.14285714285714285
0.125
0.1111111111111111
0.1
```

5. Write a program using a for loop that loops over a sequence.

```
#For loop in list
color_list=['red','green','white','purple','pink','marron']
for i in color_list:
   print(i)

#For loop in string
fr_string="Goeduhub technologies"
for i in fr_string:
    print(i)

#For loop in dictionary
fr_dict={1:'q',2:'w',3:'e',4:"r"}
for i in fr_dict:
    print(i)
    print(fr_dict[i])
```

```
#Sum of all the list elements
Num=[1,2,3,4,5,6,7,8,9,10]
Sum=0
for i in Num:
    Sum=Sum+i
print(Sum)

Output:
red
green
white
purple
pink
marron
G
o
e
d
u
h
u
b

t
e
c
h
n
o
l
o
g
i
e
s
1
q
2
w
3
e
4
r
      55
```

6. Write a program using while loop that asks the user for a number, and prints a countdown from that number to zero.

```
num=int(input("Enter your number"))
while(num>=0):
    print(num)
    num=num-1
```

Output:

```
Enter your number5
5
4
3
2
1
0
```

Experiment No.: 3

Topics to be covered : List, Tuples, Set, Dictionaries

Exercise based on above Topics:

1. Write a program to demonstrate list in python
2. Write a program to demonstrate tuples in python
3. Write a program to demonstrate Set in python
4. Write a program to demonstrate Dictionaries in python
5. Write a program to check number is Armstrong or not
6. Write a program to check number is palindrome or not
7. Write a program to check number is prime or not

1) Write a program to demonstrate list in python

```
#Create a List:
thislist = ["apple", "banana", "cherry"]
print(thislist)
Print the number of items in the list:
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
#Print the second item of the list:
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
#Return the third, fourth, and fifth item:
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon"]
print(thislist[2:5])
```

2)Write a program to demonstrate tuples in python

```
#Example
#Tuples allow duplicate values:
thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)
#Example
#Print the number of items in the tuple:
thistuple = ("apple", "banana", "cherry")
print(len(thistuple))
#Example
#One item tuple, remember the commma:
thistuple = ("apple",)
print(type(thistuple))
```

```
#NOT a tuple
thistuple = ("apple")
print(type(thistuple))
#Example
#A tuple with strings, integers and boolean values:
tuple1 = ("abc", 34, True, 40, "male")
print(tuple1)
#Example
#Print the second item in the tuple:
thistuple = ("apple", "banana", "cherry")
print(thistuple[0])
#Example
#Print the last item of the tuple:
thistuple = ("apple", "banana", "cherry")
print(thistuple[-2])
#Example
#Convert the tuple into a list to be able to change it:
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "kiwi"
x = tuple(y)
print(x)
```

3) Write a program to demonstrate Set in python

```
#Example
#Create a Set:
thisset = {"apple", "banana", "cherry"}
print(thisset)
#Example
#Get the number of items in a set:
thisset = {"apple", "banana", "cherry"}
print(len(thisset))
#Example
#String, int and boolean data types:
set1 = {"apple", "banana", "cherry"}
set2 = {1, 5, 7, 9, 3}
set3 = {True, False, False}
print(set1)
print(set2)
print(set3)
thisset = set(("apple", "banana", "cherry")) # note the double round-
brackets
print(thisset)
#Example
#Loop through the set, and print the values:
```

```
thisset = {"apple", "banana", "cherry"}
for x in thisset:
  print(x)
#Example
#Add an item to a set, using the add() method:
thisset = {"apple", "banana", "cherry"}
thisset.add("orange")
print(thisset)
```

4) Write a program to demonstrate Dictionaries in python

```
#Create and print a dictionary:
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
#Print the "brand" value of the dictionary:
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["brand"])
#There is also a method called get() that will give you the same result:
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict.get("model")
print(x)
#String, int, boolean, and list data types:
thisdict = {
  "brand": "Ford",
  "electric": False,
  "year": 1964,
  "colors": ["red", "white", "blue"]
}
print(thisdict)
#Get a list of the keys:
thisdict = {
  "brand": "Ford",
  "electric": False,
  "year": 1964,
```

```
  "colors": ["red", "white", "blue"]
}
x = thisdict.keys()
print(x)
#Example
#Get a list of the values:
car = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
x = car.values()
print(x)
#Example
#Loop through both keys and values, by using the items() method:
for x, y in thisdict.items():
  print(x, y)
#Create three dictionaries, then create one dictionary that will contain t
he other three dictionaries:
child1 = {
  "name" : "Emil",
  "year" : 2004
}
child2 = {
  "name" : "Tobias",
  "year" : 2007
}
child3 = {
  "name" : "Linus",
  "year" : 2011
}
myfamily = {
  "child1" : child1,
  "child2" : child2,
  "child3" : child3
}
print(myfamily)
```

4) Write a program to check number is Armstrong or not

```
# take input from the user
num = int(input("Enter a number: "))

# initialize sum
sum = 0
```

```
# find the sum of the cube of each digit
temp = num
while temp > 0:
   digit = temp % 10
   sum =sum+digit ** 3
   temp=temp//10
   print(temp)

# display the result
if num == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")
output:
Enter a number: 121
12
1
0
121 is not an Armstrong number
```

6) Write a program to check number is palindrome or not

```
num = int(input("Enter a value:"))
temp = num
rev = 0
while(num > 0):
    dig = num % 10
    rev = rev * 10 + dig
    num = num // 10
if(temp == rev):
    print("This value is a palindrome number!")
else:
    print("This value is not a palindrome number!")

output:
Enter a value:121
This value is a palindrome number!
```

7) Write a program to check number is prime or not

```
# To take input from the user
num = int(input("Enter a number: "))

# define a flag variable
flag = False

# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2, num):
        #print(i)
        if (num % i) == 0:
            # if factor is found, set flag to True
            flag = True
            # break out of loop
            break

# check if flag is True
if flag:
    print(num, "is not a prime number")
else:
    print(num, "is a prime number")

Output:
Enter a number: 11
11 is a prime number
```

Experiment No.: 4

Topics to be covered : Strings, Array, File Handling

Exercise based on above Topics:

1. Write a program to count the numbers of characters in the string and store them in a dictionary data structure
2. Write a program to use split and join methods in the string and trace a birthday of a person with a dictionary data structure
3. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?
4. Write a program to print each line of a file in reverse order.
5. Write a program to compute the number of characters, words and lines in a file.
6. Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on.
7. Write a program to implement Insertion sort.
8. Write a program to implement Selection sort.

1. Write a program to count the numbers of characters in the string and store them in a dictionary data structure

```
str=input("enter string : ")

f = {}

for i in str:
    #print (i)

    if i in f:

        f[i] += 1

    else:

        f[i] = 1

print(f)

output:
enter string : shikha
```

```
{'s': 1, 'h': 2, 'i': 1, 'k': 1, 'a': 1}
```

2. Write a program to use split and join methods in the string and trace a birthday of a person with a dictionary data structure

```
bdaystr=input("Enter date of birth:\n")

bdaylist=bdaystr.split("/")
print(bdaylist)

bday='-'.join(bdaylist)

bdaydict={"birthday":bday}

print(bdaydict['birthday'])

req= input("enter bday to match")

if bdaydict['birthday']==req:

  print("yes,Birthday date matched")

else:

  print("no,Birthday date not matched")
output:
Enter date of birth:
22/4/2020
['22', '4', '2020']
22-4-2020
enter bday to match23-4-2020
no,Birthday date not matched
```

3. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

```
f="D:\\myfile\demofile.txt"

file = open ( f, "r" )

a=[]
b={}
for i in file:
```

```
    for j in range(0,len(i)):

        a.append(i[j])
    print(a)

for i in a:

    if i in b:

        b[i]+=1

    else:

        b[i]=1

print(b)


c=f.split(".")

if c[1]=="txt":

    print("\n\nit is a text file")

elif c[1]=="cpp":

    print("\n\nit is a c++ file")

else:

    print("\n\nit is a c file")
```

output:

```
['W', 'o', 'o', 'p', 's', '!', ' ', 'I', ' ', 'h', 'a', 'v', 'e', ' ', 'd',
'e', 'l', 'e', 't', 'e', 'd', ' ', 't', 'h', 'e', ' ', 'c', 'o', 'n', 't',
'e', 'n', 't', '!']
{'W': 1, 'o': 3, 'p': 1, 's': 1, '!': 2, ' ': 5, 'I': 1, 'h': 2, 'a': 1, 'v':
1, 'e': 6, 'd': 2, 'l': 1, 't': 4, 'c': 1, 'n': 2}


it is a text file
```

4. Write a program to print each line of a file in reverse order.

```
ofile=open("D:\\myfile\demofile.txt","r")
k=ofile.readlines()
t=reversed(k)
for i in t:
 print(i.rstrip())
ofile.close()
```

output:

```
we are performing experiments
 we are performing experiments
 we are performing experiments
file handling concepts we are performing experiments we are performing
experiments
this is my first file
world
hello
```

5. Write a program to compute the number of characters, words and lines in file.

```
file = open("D:\\myfile\demofile.txt","r")

number_of_lines = 0
number_of_words = 0
number_of_characters = 0
for line in file:
  line = line.strip()


  words = line.split()
  number_of_lines += 1
  number_of_words += len(words)
  number_of_characters += len(line)

file.close()
print("lines:", number_of_lines, "words:", number_of_words,
"characters:", number_of_characters)
```

output:

```
lines: 7 words: 30 characters: 200
```

6. Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on.

```
string1 = input("Enter first string: ")

string2 = input("Enter second string: ")

if string1 == string2:

    print("\nBoth strings are equal to each other.")

    print(string1,"==",string2);

else:

    print("\nStrings are not equal.")

    print(string1,"!=",string2);
```

```
output:
Enter first string: shikha
Enter second string: shikha

Both strings are equal to each other.
shikha == shikha
```

7. Write a program to implement Insertion sort.

```
a = [16, 19, 11, 15, 10, 12, 14]

#iterating over a
for i in a:
    j = a.index(i)
    #i is not the first element
    while j>0:
        #not in order
        if a[j-1] > a[j]:
            #swap
            a[j-1],a[j] = a[j],a[j-1]
        else:
            #in order
            break
        j = j-1
```

```
print (a)

output:
       [10, 11, 12, 14, 15, 16, 19]
```

8. Write a program to implement Selection sort.

```
a = [16, 19, 11, 15, 10, 12, 14]

i = 0
while i<len(a):
    #smallest element in the sublist
    smallest = min(a[i:])
    #index of smallest element
    index_of_smallest = a.index(smallest)
    #swapping
    a[i],a[index_of_smallest] = a[index_of_smallest],a[i]
    i=i+1
print (a)

output:
       [10, 11, 12, 14, 15, 16, 19]
```