



Established in Year 2000 - RTU QIV RANKED A

RIET

RAJASTHAN INSTITUTE OF
ENGINEERING AND TECHNOLOGY

Approved by AICTE and Affiliated to Rajasthan Technical University, Kota.



PYTHON LAB MANUAL

Lab Code:- 6CS4-23

(III Year B.Tech. VI Sem)

Session 2020-21



Department of Computer Science & Engineering

RIET, JAIPUR

Submitted to:

MR.DINESH SWAMI
ASSISTANT PROFESSOR

Submitted by:

Vicky Kumar
18ERECS080 (CSE)
Sem-6th



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)

OL+OT+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

Experiment 1

Aim: Write a program to demonstrate basic data type in python.

Code:

```
a = 5
print("Type of a: ", type(a))
b = 5.0
print("\nType of b: ", type(b))
c = 2 + 4j
print("\nType of c: ", type(c))
```

Creating a String

```
String1 = 'Welcome to the RIET Jaipur '
print("String with the use of Single Quotes: ")
print(String1)
```

Creating a List with

```
List = ["RIET", "JAIPUR", "RAJ"]
print("\n List containing multiple values: ")
print(List[0])
print(List[2])
```

Creating a Set set1 = set()

```
set1.add(8) # Creating a Tuple with
the use of list list1 = [1, 2, 4, 5, 6]
print("\nTuple using List: ")
print(tuple(list1))
set1.add(9)
set1.add((6, 7))
print("\nSet after Addition of Three elements: ")
print(set1)
```

Output:

```
C:\Users\Vicky\Desktop\clg pdf\College PDFs\semester 6th\python lab\lab record>python "experiment 1.py"
int operations
<class 'int'>
1.5
<class 'float'>
1
<class 'int'>
=====
<class 'float'>
3.0
<class 'float'>
=====
len of string is 12
To upper  HELLO WORLD!
To Lower  hello world!
To Title  Hello World!
Get index of 'll' in string  2
=====
<class 'complex'>
Imaginary :  6.0
Real :  1.0
Cojugate :  (1-6j)
=====
lst1 [1, 2]
lst2 [3, 4]
len of lst1 2
lst1 [1]
lst2 [4]
[]
lst2 deleted
=====
tuple (1,)
1
=====
dict_keys([1, 2, 3])
dict_values([5, 3, 8])
1 = 5
2 = 3
3 = 8
{2: 3, 3: 8}
```

Experiment 2

Aim: (a) Write a program to compute distance between two points taking input from the user. **Code:**

```
import math
a=int(input("Enter first value"))
b=int(input("Enter second value"))
c=math.sqrt(a**2+b**2)
print("Distance=",c)
```

Output:

Enter first value5

Enter second value6

Distance= 7.810249675906654

distance formula

$$D = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$$

Enter x1 and y1 : 7 9

Enter x2 and y2 : 4 3

Distance between point1 (7, 9) and point2 (4, 3) is 6.71 u

Aim: (b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

Code:

```
import sys
a=int(sys.argv[1])
b=int(sys.argv[2])
c=a+b
print("Sum=",c)
```

Output:

```
python add.py 4 5
Sum= 9
```

Experiment 3

Aim: (a) Write a Program for checking whether the given number is an even number or not. Code:

```
num = int(input("Enter a number: "))
if(num%2==0):
    print("This is an even number.")
else:
    print("This is an odd number.")
```

Output:

Enter a number: 4

This is an even number.

```
1 def odd_even(num: int):
2     if num&1:
3         return "odd"
4     return "even"
5
6 print(odd_even(8))
7 print(odd_even(21))
8 print(odd_even(0))
9
```

```
even
odd
even
```

Aim: (b) Using a for loop, write a program that prints out the decimal equivalents of $1/2$, $1/3$, $1/4$, . . . , $1/10$.

Code:

```
for i in range(1,11):  
    print ("Decimal equivalent value for 1/",i," is",1/float(i))
```

Output:

```
1  def frac_n(n):  
2      for i in range(2, n+1):  
3          print('{:<5} = {:>10.2f}'.format('1/' + str(i), 1/i))  
4  
5  frac_n(15)
```

```
1/2   =      0.50  
1/3   =      0.33  
1/4   =      0.25  
1/5   =      0.20  
1/6   =      0.17  
1/7   =      0.14  
1/8   =      0.12  
1/9   =      0.11  
1/10  =      0.10  
1/11  =      0.09  
1/12  =      0.08  
1/13  =      0.08  
1/14  =      0.07  
1/15  =      0.07
```


Experiment 4

Aim: (a) Write a Program to demonstrate list in python (We are given an array of n distinct numbers, the task is to sort all even-placed numbers in increasing and odd-placed numbers in decreasing order. The modified array should contain all sorted even-placed numbers followed by reverse sorted odd-placed numbers.)

Code:

```
def evenOddSort(input):
```

```
    # separate even odd indexed elements list
```

```
    evens = [ input[i] for i in range(0,len(input)) if i%2==0 ]
```

```
    odds = [ input[i] for i in range(0,len(input)) if i%2!=0 ]
```

```
    # sort evens in ascending and odds in descending using sorted() method
```

```
    print (sorted(evens) + sorted(odds,reverse=True))
```

```
input = [0, 1, 2, 3, 4, 5, 6, 7]
```

```
evenOddSort(input)
```

Output:

```
[0, 2, 4, 6, 7, 5, 3, 1]
```

```
original array [1, 6, 5, 2, 4, 0, 3]
```

```
sorted array [1, 6, 3, 2, 4, 0, 5]
```

Aim: (b) Write a Program to demonstrate tuple in python (Given a list of tuples, Write a Python program to remove all the duplicated tuples from the given list).

Code:

```
def removeDuplicates(lst):  
    return [t for t in (set(tuple(i) for i in lst))]
```

Driver code

```
lst = [(1, 2), (5, 7), (3, 6), (1, 2)]  
print(removeDuplicates(lst))
```

Output:

```
[(1, 2), (5, 7), (3, 6)]
```

```
Original List of Tuples           : [(1, 2), (5, 7), (3, 6), (1, 2)]  
After deleting Duplicates tuples from list : [(5, 7), (3, 6), (1, 2)]
```

Aim: (c) Write a program using a for loop that loops over a sequence.

Code:

```
players=["kohli", "dhoni", "sachin", "sehwag", "Dravid"]  
for i in players:  
    print (i)
```

Output:

```
kohli  
dhoni  
sachin  
sehwag  
Dravid
```

Aim: (d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

Code:

```
n=int(input("Enter the number for countdown: "))
while (0<=n):
    print (n, end=" ")
    n=n-1
```

Output:

```
Enter the number for countdown: 15
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
```

Experiment 5

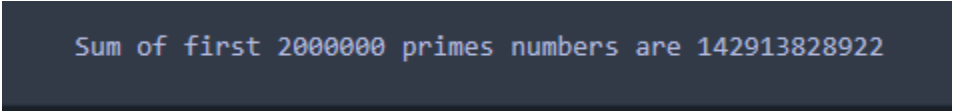
Aim: (a) Find the sum of all the primes below two million.

Code:

```
n = 2000000
prime = [True for i in range(n+1)]
p = 2
while (p * p <= n):
    if (prime[p] == True):
        for i in range(p * p, n+1, p):
            prime[i] = False
        p += 1
sum=0
for p in range(2, n):
    if prime[p]:
        sum=sum+p
print("sum=", sum)
```

Output:

sum= 142913828922



```
Sum of first 2000000 primes numbers are 142913828922
```

Aim: (b) 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.

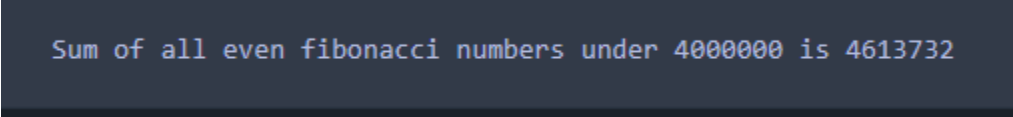
Code:

```
limit = 4000000
if (limit < 2):
    print("Sum=0")

else:
    ef1 = 0
    ef2 = 2
    sm= ef1 + ef2

    while (ef2 <= limit):
        ef3 = 4 * ef2 + ef1
        if (ef3 > limit):
            break
        ef1 = ef2
        ef2 = ef3
        sm = sm + ef2
    print("Sum=",sm)
```

Output: Sum= 4613732



Sum of all even fibonacci numbers under 4000000 is 4613732

Experiment 6

Aim: (a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure.

Code:

```
def char_frequency(str1):
    dict = {}
    for n in str1:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict
print(char_frequency('google.com'))
```

Output:

```
{'c': 1, 'e': 1, 'g': 2, 'm': 1, 'l': 1, 'o': 3, '.': 1}
```

```
{'M': 1, 'y': 2, ' ': 4, 'n': 1, 'a': 2, 'm': 2, 'e': 1, 'i': 2, 's': 1, 'v': 1, 'c': 1, 'k': 1, 'K': 1, 'u': 1, 'r': 1}
```

Aim: (b) Write a program to use split and join methods in the string and trace a birthday of a person with a dictionary data structure.

Code:

```
dob={"mothi":"12-11-1990","sudheer":"17-08-1991","vinay":"31-08-1988"}
str1=input("which person dob you want: ")
l=str1.split()
birth=""
for i in l:
    if i in dob.keys():
        name=i
print (" ".join([name,"Birthday is",dob[name]]))
```

Output:

```
which person dob you want: i want vinay dob
vinay Birthday is 31-08-1988
```

Experiment 7

Aim: 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?

Code:

```
import os
f=open("deepa.py")
count=dict()
for line in f:
    for ch in line:
        if ch in count:
            count[ch]=count[ch]+1
        else:
            count[ch]=1
print (count)
filename,file_extension=os.path.splitext("deepa.py");
print("file_extension==",file_extension);
if(file_extension=='.py'):
    print("its python program file");
elif(file_extension==".txt"):
    print("its a txt file");
elif(file_extension=="c"):
    print("its a c program file");
f.close()
```

deepa.py:
my name is deepa modi

Output:

```
{'m': 3, 'y': 1, ' ': 4, 'n': 1, 'a': 2, 'e': 3, 'i': 2, 's': 1, 'd': 2, 'p': 1, 'o': 1}
file_extension== .py
its python program file
```

```
28 check_file("C:/Users/ChandraKumawat/Desktop/test.py")
```

```
{'u': 15, 's': 8, 'i': 29, 'n': 32, 'g': 12, ' ': 201, 'a': 40, 'm': 2, 'e': 29, 'p': 12, 'c': 16, 't': 36, 'd': 50, ';': 19, '\n': 39,
'/' : 10, 'A': 3, 'l': 4, 'y': 4, 'f': 7, 'o': 18, 'r': 21, 'h': 8, '.' : 3, 'v': 15, 'E': 8, '(' : 16, '<': 14, '>': 4, 'j': 16, '[' : 6,
']': 6, ',': 18, ')': 16, '{': 4, '_' : 2, 'b': 2, 'k': 2, '}' : 4, 'G': 2, 'V': 5, '=' : 2, '0': 4, '+': 2, '"' : 8, '\\': 3, 'x': 3, ':': 1,
'-': 1, 'D': 1, '5': 1, '1': 4, '4': 3, '2': 2, '3': 3}
file_extension== .py
its python program file
```

Experiment 8

Aim: (a) Write a program to print each line of a file in reverse order. Code:

```
filename=input("Enter the filename: ")
f=open(filename,"r")
for line in f:
    line2=""
    for ch in range(len(line)-1,-1,-1):
        line2=line2+line[ch]
    print(line2)
f.close()
```

deepa.py:
my name is deepa modi
i am a cool person

Output:
Enter the filename: deepa.py
idom apeed si eman ym
nosrep looc a ma i

Aim: (b) Write a program to compute the number of characters, words and lines in a file. Code:

```
filename=input("Enter the filename: ")
f=open(filename,"r")
l=w=c=0
for line in f:
    words=line.split()
    l=l+1
    for word in words:
        w=w+1
        for ch in word:
            c=c+1
print("No. of lines",l)
print("No. of words",w)
print("No. of characters",c)
f.close()
```

deepa.py:
my name is deepa modi
i am a cool person

Output:
Enter the filename:
deepa.py

No. of lines 2
No. of words 10
No. of characters 31

Experiment 9

Aim: (a) 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.

Code:

```
def mutate(word):
    out_list = []
    letters = 'abcdefghijklmnopqrstuvwxyz'
    #insert a character
    for i in range(len(word) + 1):
        for j in range(26):
            out_list.append(word[:i] + letters[j] + word[i:])
    #deleting a character
    for i in range(len(word)):
        out_list.append(word[:i] + word[i + 1:])
    #replace a character
    for i in range(len(word)):
        for j in range(26):
            out_list.append(word[:i] + letters[j] + word[i + 1:])
    #swapping a characters
    current_word = []
    out_word = ""
    for i in range(len(word) - 1):
        for j in range(i + 1, len(word)):
            #converting string into list
            cword = list(word)
            #Swapping of characters in a list
            cword[i], cword[j] = cword[j], cword[i]
            #converting list into string
            str1="".join(current_word)
            out_list.append(str1)
    return out_list

def nearly_equal(word1, word2):
    if len(word1)<len(word2):
        word1,word2=word2,word1
    return word1 in mutate(word2)
    else:
        return word1 in mutate(word2)

a=input("Enter First Word: ")
b=input("Enter Second Word: ")
print(nearly_equal(a,b))
```

Output:

Enter First Word: deepa
Enter Second Word: dipa
False

Enter First Word: welcome
Enter Second Word: welcme
True

```
Enter test cases : 2
Enter First Word: Nileesh
Enter Second Word: Niteesh
True
Enter First Word: Nakul
Enter Second Word: Naakull
False
```

Aim: (b) Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line. Code:

```
def gcd(x,y):
    return x if y==0 else gcd(y,x%y)
def lcm(x,y):
    return (x*y)//gcd(x,y)
print ("gcd is",gcd(54,24))
print ("lcm is",lcm(54,24))
```

Output:

gcd is 6
lcm is 216

```
Enter space separated numbers a and b33 98
GCD of a 33 b, 98 is 1
LCM of a 33 b, 98 is 3234
```

Experiment 10

Aim: (a) Write a program to implement Merge sort. Code:

```
def mergeSort(nlist):
    #print("Splitting ",nlist)
    if len(nlist)>1:
        mid = len(nlist)//2
        lefthalf = nlist[:mid]
        righthalf = nlist[mid:]

        mergeSort(lefthalf)
        mergeSort(righthalf)
        i=j=k=0
        while i < len(lefthalf) and j < len(righthalf):
            if lefthalf[i] < righthalf[j]:
                nlist[k]=lefthalf[i]
                i=i+1
            else:
                nlist[k]=righthalf[j]
                j=j+1
            k=k+1

        while i < len(lefthalf):
            nlist[k]=lefthalf[i]
            i=i+1
            k=k+1

        while j < len(righthalf):
            nlist[k]=righthalf[j]
            j=j+1
            k=k+1
    #print("Merging ",nlist)

nlist = [14,46,43,27,57,41,45,21,70]
mergeSort(nlist)
print(nlist)
```

Output:

[14, 21, 27, 41, 43, 45, 46, 57, 70]

Aim: (b) Write a program to implement Selection sort. Code:

30

```
def selectionSort(nlist):
    for fillslot in range(len(nlist)-1,0,-1):
        maxpos=0
        for location in range(1,fillslot+1):
            if nlist[location]>nlist[maxpos]:
                maxpos = location

        temp = nlist[fillslot]
        nlist[fillslot] = nlist[maxpos]
        nlist[maxpos] = temp

nlist = [14,46,43,27,57,41,45,21,70]
selectionSort(nlist)
print(nlist)
```

Output:

[14, 21, 27, 41, 43, 45, 46, 57, 70]

Aim: (c) Write a program to implement Insertion sort. Code:

```
def insertionSort(nlist):
    for index in range(1,len(nlist)):

        currentvalue = nlist[index]
        position = index

        while position>0 and
            nlist[position-1]>currentvalue:
                nlist[position]=nlist[position-1]
                position = position-1

        nlist[position]=currentvalue

nlist = [14,46,43,27,57,41,45,21,70]
insertionSort(nlist)
print(nlist)
```

Output:

[14, 21, 27, 41, 43, 45, 46, 57, 70]