

BASIC PYTHON PROGRAMS

Lab 1(a)

```
x = int(input("enter integer value \n"))
print(type(x))
print(x)
```

```
x = float(input("enter float value \n"))
print(type(x))
print(x)
```

```
x = str(input("enter a string \n"))
print(type(x))
print(x)
```

```
a = bool(int(input("Enter boolean value \n ")))
print(type(a))
print(a)
```

```
l = input("Enter three value with space \n").split()
print(type(l))
print(l)
```

```
t = input("\n Enter 3 elements with space
\n").split()
x = tuple(t)
print(type(x),x)
```

```
print("\n Enter dictionary values ")
d = {}
for i in range(3) :
    K,V = input("Enter key and value ").split()
    d.update({K:V})

print(type(d),d)
```

Output 1(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\1a.py"
enter integer value
5
<class 'int'>
5
enter float value
4.5
<class 'float'>
4.5
enter a string
python
<class 'str'>
python
Enter boolean value
1
<class 'bool'>
True
Enter three value with space
4 5 6
<class 'list'>
['4', '5', '6']

Enter 3 elements with space
1 2 3
<class 'tuple'> ('1', '2', '3')

Enter dictionary values
Enter key and value 1 11
Enter key and value 2 22
Enter key and value 3 33
<class 'dict'> {'1': '11', '2': '22', '3': '33'}
PS E:\4TH SEM\SK Python> █
```

Lab 1(b)

```
print("mathematical operations ")

a=float(input("enter a = "))
b=float(input("enter b = "))
print("\n add = ",a+b,"\n sub = ",a-b,"\n mul = ",a*b,
"\n div = ",a/b,"\n mod = ",a%b,"\n power = ",a**b)

print("\n")

c=float(input("enter c = "))
d=float(input("enter d = "))

e=(a+b)*c/d
print("value of (a*b)*c/d is ",e)

e=((a+b)*c)/d
print("value of ((a+b)*c)/d is ",e)

e=(a+b)*(c/d)
print("value of (a+b)*(c/d) is ",e)

e=a+(b*c)/d
print("value of a+(b*c)/d is ",e)
```

Output 1(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\1b.py"
mathematical operations
enter a = 5
enter b = 4

add = 9.0
sub = 1.0
mul = 20.0
div = 1.25
mod = 1.0
power = 625.0

enter c = 4.5
enter d = 8.2
value of (a*b)*c/d is 4.939024390243903
value of ((a+b)*c)/d is 4.939024390243903
value of (a+b)*(c/d) is 4.939024390243903
value of a+(b*c)/d is 7.195121951219512
PS E:\4TH SEM\SK Python> 
```

Lab 2(a)

```
import math

n = int (input("Enter number of terms \n"))
sum = 0
avg = 0

for i in range(1,n+1):
    sum = sum+i

print("sum is" , sum)
avg = sum/n
print("avg is ", avg)
```

Output 2(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\2a.py"
Enter number of terms
5
sum is 15
avg is 3.0
PS E:\4TH SEM\SK Python> █
```

Lab 2(b)

Factorial using Iteration

```
num = int(input("enter a value \n"))

factorial = 1

if num < 0:
    print("factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

Factorial using Recursion

```
def factorial(x):
    if x == 1:
        return 1
    else:
        return (x * factorial(x-1))

num = int(input("Enter a value \n"))
result = factorial(num)
print("The factorial of", num, "is", result)
```

Fibonacci Using Iteration

```
nterms = int(input("enter number of terms? "))
```

```
n1 = 0
```

```
n2 = 1
```

```
count = 0
```

```
if nterms <= 0:
```

```
    print("Please enter a positive integer")
```

```
elif nterms == 1:
```

```
    print("Fibonacci sequence upto",nterms,":")
```

```
    print(n1)
```

```
else:
```

```
    print("Fibonacci sequence:")
```

```
    while count < nterms:
```

```
        print(n1)
```

```
        nth = n1 + n2
```

```
        n1 = n2
```

```
        n2 = nth
```

```
        count += 1
```

Fibonacci Using Recursion

```
def recur_fibo(n):  
    if n <= 1:  
        return n  
    else:  
        return(recur_fibo(n-1) + recur_fibo(n-2))  
  
terms = int(input("enter a value\n"))  
  
if terms <= 0:  
    print("Plese enter a positive integer")  
else:  
    print("Fibonacci sequence:")  
    for i in range(terms):  
        print(recur_fibo(i))
```


Output 2(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\2b.py"
enter a value
6
The factorial of 6 is 720
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
Enter a value
5
The factorial of 5 is 120
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
File.py"
enter number of terms? 7
Fibonacci sequence:
0
1
1
2
3
5
8
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
enter a value
9
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
PS E:\4TH SEM\SK Python> 
```

Lab 3(a)

```
low=int(input("enter lower value : "))
uppr=int(input("enter upper value: "))
print("\nprime and odd between lower and upper limit
are : ")
for i in range(low,uppr+1):
    if i>1 and ( i%2!=0):
        for j in range(2,i):
            if(i%j)==0:
                break
        else:
            print(i)
```

Output 3(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\3a.py"
Enter lower number : 0
Enter upper number : 30
Prime numbers between 0 and 30 are:
2
3
5
7
11
13
17
19
23
29
PS E:\4TH SEM\SK Python> █
```

Lab 3(b)

```
a = int(input('Enter first number : '))
b = int(input('Enter second number : '))
c = int(input('Enter third number : '))

smallest = 0
greatest = 0

if a < b and a < c :
    smallest = a
elif b < a and b < c :
    smallest = b
else :
    smallest = c

if a > b and a > c :
    greatest = a
elif b > a and b > c :
    greatest = b
else :
    greatest = c

print("Smallest no is " , smallest)
print("greatest no is " , greatest)
```

Output 3(b)

```
Enter first number : 7
Enter second number : 8
Enter third number : 1
Smallest no is 1
greatest no is 8
PS E:\4TH SEM\SK Python>
```

Lab 3(c)

```
num1=int(input("enter the first number: "))
num2=int(input("enter the second number: "))
if(num1>num2):
    smaller=num1
else:
    smaller=num2
for i in range (1,smaller+1):
    if(num1%i==0)and(num2%i==0):
        hcf=i
        lcm=(num1*num2)/hcf
print("the hcf of two numbers is :",hcf)
print("the lcm of two numbers is :",lcm)
```

Output 3(c)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH S
EM\SK Python\3c.py"
enter the first number: 15
enter the second number: 25
the hcf of two numbers is : 5
the lcm of two numbers is : 75.0
PS E:\4TH SEM\SK Python> █
```

Lab 4(a)

```
import itertools , random
deck =
list(itertools.product(range(1,14),['spade','heart','diamond',
'club']))
random.shuffle(deck)
for i in range(8):
    print(deck[i][0],"of",deck[i][1])
```

Output 4(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\4a.py"
5 of heart
13 of spade
5 of diamond
11 of club
6 of heart
13 of heart
7 of spade
4 of spade
PS E:\4TH SEM\SK Python>
```

Lab 4(b)

```
import calendar

year = int(input("Enter year: "))
month = int(input("Enter month: "))
print(calendar.month(year, month))

# full calendar
print(calendar.calendar(2022))
```

Output 4(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
Enter year: 2023
Enter month: 9
    September 2023
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

PS E:\4TH SEM\SK Python> █
```

Lab 4(c)

```
from datetime import date , datetime, time

print("current date and time ", datetime.now())

my_date = date(2022,2,27)

print("Date passed as user is", my_date)

print("Today is " ,date.today())

Today = date.today()
```

Output 4(c)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
current date and time  2023-06-02 15:49:26.432416
Date passed as user is 2022-02-27
Today is  2023-06-02
PS E:\4TH SEM\SK Python>
```

Lab 5(a)

```
s = input ("Enter a string: ")
for i in s:
    print(i)
print(s[1:3])
s1 = input("Enter a string 1:")
s2 = input ("Enter a string 2:")
s3 = s1 + s2
print(s3)
list1 = ['JACK', 'CLUB', 'HEART', 'SPADE']
list1.sort()
print(list1)
```

Output 5(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\5a.py"
Enter a string: Programming
P
r
o
g
r
a
m
m
i
n
g
ro
Enter a string 1: Python
Enter a string 2: Lab
Python Lab
['CLUB', 'HEART', 'JACK', 'SPADE']
PS E:\4TH SEM\SK Python> █
```


Lab 5(b)

palindrome check

```
s = input("Enter a string to check the paindrome: ")
```

```
res = True
```

```
n = len(s)
```

```
print(n)
```

```
for i in range(int (n/2)):
```

```
    if(s[i] != s[n -1 -i]):
```

```
        res = False
```

```
print(res)
```

#punctuation

```
string = input("Enter a bunch of strings with punctuations  
in between : ")
```

```
print("Before removing punctuation : ",string)
```

```
punctuation = ' ' ! @ # $ % ^ & * ( ) _ - = + [ ] { } ; : '  
" \ | , < . > / ? '''
```

```
for i in string :
```

```
    if i in punctuation :
```

```
        string = string.replace(i," ")
```

```
print(string)
```

Output 5(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\5b.py"
```

```
Enter a string to check the paindrome: radar
```

```
5
```

```
True
```

```
Enter a bunch of strings with punctuations in between : You # are #@#% *(a!@# good&:"< student
```

```
Before removing punctuation : You # are #@#% *(a!@# good&:"< student
```

```
You are a good student
```

```
PS E:\4TH SEM\SK Python> █
```

Lab 6(a)

```
set1 = ("btr",11,12,13)
```

```
l1 = list(set1)
```

```
print(type(l1))
```

```
for i in l1:
```

```
    print(i)
```

```
print(l1[0:2])
```

```
l1.append("ise")
```

```
print(l1)
```

```
l1.insert(3,888)
```

```
print(l1)
```

```
l1.remove(888)
```

```
print(l1)
```

Output 6(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH S
EM\SK Python\6a.py"
<class 'list'>
btr
11
12
13
['btr', 11]
['btr', 11, 12, 13, 'ise']
['btr', 11, 12, 888, 13, 'ise']
['btr', 11, 12, 13, 'ise']
PS E:\4TH SEM\SK Python>
```

Lab 6(b)

```
t1 = (15,90,105,120,135,150)
x = tuple(t1)
print(type(x))

for i in x:
    print(i)

print(x[3:6])

x = x + ("jain","university",2023)
print(x)

l1 = list(x)
print(type(l1))
l1.insert(3,"Python")
print(l1)

l1.remove(105)
l1.remove(120)
l1.remove(135)

x = tuple(l1)
print(type(x))
print(x)
```

Output 6(b)

```
EM\SK Python\tempCodeRunnerFile.py"
<class 'tuple'>
15
90
105
120
135
150
(120, 135, 150)
(15, 90, 105, 120, 135, 150, 'jain', 'university', 2023)
<class 'list'>
[15, 90, 105, 'Python', 120, 135, 150, 'jain', 'university', 2023]
<class 'tuple'>
(15, 90, 'Python', 150, 'jain', 'university', 2023)
PS E:\4TH SEM\SK Python>
```

Lab 6(c)

```
import itertools

d1 = {"one" : 10, "two" : 20 , "three" : 30 , "four" :40 }
print(type(d1))
print(d1)

for i in d1.keys() :
    print(i)

for j in d1.values() :
    print(j)

for i, j in d1.items() :
    print(i ,":",j)

# slicing using itertools

newd1 = dict(itertools.islice(d1.items(), 1,3))
print(newd1)

# adding
d1.update(five = 50 , six = 60)
print(d1)

# removing
d1.pop("two")
d1.pop("three")
print(d1)
```

Output 6(c)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\6cc.py"
<class 'dict'>
{'one': 10, 'two': 20, 'three': 30, 'four': 40}
one
two
three
four
10
20
30
40
one : 10
two : 20
three : 30
four : 40
{'two': 20, 'three': 30}
{'one': 10, 'two': 20, 'three': 30, 'four': 40, 'five': 50, 'six': 60}
{'one': 10, 'four': 40, 'five': 50, 'six': 60}
PS E:\4TH SEM\SK Python>
```

Lab 7(a)

```
import numpy as np

a1 = np.array(5)
print(type(a1))
print(a1)
print("dim of a1 ", a1.ndim)

a2 = np.array([5,6,7,8,9])
print(type(a2))
print(a2)
print("dim of a1 ", a2.ndim)

b1 = [2,3]
b2 = [4,5]
a3 = np.array([b1,b2])
print(a3, type(a3))
print("dim of a3 ", a3.ndim )

a4 = np.array([[b1],[b2]])
print(a4, type(a3))
print("dim of a3 ", a4.ndim )

a5 = np.array([[[b1],[b2]]])
print(a5, type(a5))
print("dim of a5", a5.ndim)
```

Output 7(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\7a.py"
<class 'numpy.ndarray'>
5
dim of a1  0
<class 'numpy.ndarray'>
[5 6 7 8 9]
dim of a1  1
[[2 3]
 [4 5]] <class 'numpy.ndarray'>
dim of a3  2
[[[2 3]]

 [[4 5]]] <class 'numpy.ndarray'>
dim of a3  3
[[[[2 3]]

 [[4 5]]]] <class 'numpy.ndarray'>
dim of a5  4
PS E:\4TH SEM\SK Python>
```

Lab 7(b)

```
import numpy as np

ar = np.array((1,2,3,4,5,6))
print(ar[0])
print(ar[1])
print(ar[-1])
print(ar[-2])
print(ar[-6])

print(ar[2]+ar[3])
print(ar[2]*ar[3])
print(ar[2]/ar[3])

ar2 = np.array([[6,5,4,3] , [7,8,9,0]])
print(" 1st row 2nd element is ", ar2[0,1])
print(" 2nd row 3rd element is ", ar2[1,2])
```

Output 7(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\7b.py"
1
2
6
5
1
7
12
0.75
 1st row 2nd element is  5
 2nd row 3rd element is  9
PS E:\4TH SEM\SK Python> █
```

Lab 7(c,d)

```
import numpy as np

m1 = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(m1.ndim)
print(m1)
m2 = np.array([[11,1,5],[5,9,7],[3,0,2]])
print(m2.ndim)
print(m2)

print(np.add(m1,m2))
print(np.subtract(m1,m2))
print(np.dot(m1,m2))
print(m2.transpose())
```

Output 7(c,d)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\7c.py"
2
[[1 2 3]
 [4 5 6]
 [7 8 9]]
2
[[11  1  5]
 [ 5  9  7]
 [ 3  0  2]]
[[12  3  8]
 [ 9 14 13]
 [10  8 11]]
[[-10  1 -2]
 [ -1 -4 -1]
 [  4  8  7]]
[[ 30 19 25]
 [ 87 49 67]
 [144 79 109]]
[[11  5  3]
 [ 1  9  0]
 [ 5  7  2]]
PS E:\4TH SEM\SK Python>
```


Lab 8(a)

```
import pandas as pd
data=[1,2,3,4]
ser=pd.Series(data)
print(ser)
```

```
d1 = {'Name' : ['Satyam' , 'Abhishek', 'Krushal', 'Neel'],
      'Age' : [17, 15, 19 ,21]}
s1 = pd.DataFrame(d1)
print(s1)
```

Output 8(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\8a.py"
0    1
1    2
2    3
3    4
dtype: int64
      Name  Age
0  Satyam   17
1 Abhishek   15
2  Krushal   19
3    Neel   21
PS E:\4TH SEM\SK Python>
```

Lab 8(b)

```
import pandas as pd
import numpy as np

d1 = {'Name' : ['Satyam' , 'Abhishek', np.NaN, 'Neel'],
      'Age' : [17, np.NaN, 19 ,21]}
s1 = pd.DataFrame(d1)
print(s1)

count = s1.isnull().sum()
print(count)
```

Output 8(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\8b.py"
   Name  Age
0  Satyam  17.0
1  Abhishek  NaN
2      NaN  19.0
3    Neel  21.0
Name      1
Age       1
dtype: int64
PS E:\4TH SEM\SK Python>
```

Lab 8(c)

```
import pandas as pd
import numpy as np

data = {"A": ["TeamA", "TeamB", "TeamB", "TeamC", "TeamA"],
        "B": [50, 40, 40, 30, 50],
        "C": [True, False, False, False, True]}
df = pd.DataFrame(data)
print(df)
print(df.drop_duplicates())
```

Output 8(c)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK
    A    B    C
0  TeamA  50  True
1  TeamB  40  False
2  TeamB  40  False
3  TeamC  30  False
4  TeamA  50  True
    A    B    C
0  TeamA  50  True
1  TeamB  40  False
3  TeamC  30  False
```

Lab 9(a)

```
file_name = "example.txt"
content = "Hello, world!"
with open(file_name, "w") as file:
    file.write(content)
print(f"File '{file_name}' has been created with content: '{content}'")
```

```
# Opening a File in Write Mode
file = open("example.txt", "w")
```

```
# Writing Content into the File
file.write("Hello, world!")
```

```
# Closing the File
file.close()
```

```
# Opening the File in Read Mode
file = open("example.txt", "r")
```

```
# Reading Content from the File
content = file.read()
print(content)
```

```
# Closing the File
file.close()
```

Output 9(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\9a.py"
File 'example.txt' has been created with content: 'Hello, world!'
Hello, world!
PS E:\4TH SEM\SK Python>
```

Lab 9(b)

```
import os
os.mkdir('folder 1')
os.rename('folder 1','folder 2')
print(os.listdir('folder 2'))

os.chdir('folder 2')
print(os.getcwd())
```

Output 9(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
['doc1.docx', 'hii.txt']
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
E:\4TH SEM\SK Python\folder 2
PS E:\4TH SEM\SK Python>
```

Lab 9(c)

```
try:
    x=1/0
except:
    print("You cant divide by zero")
finally:
    print("The try except block has completed")
```

Output 9(c)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH S
EM\SK Python\9c new.py"
You cant divide by zero
The try except block has completed
PS E:\4TH SEM\SK Python>
```

Lab 10(a)

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt

os.chdir("C:/Users/SATYAM KUMAR/Desktop")
dataset = pd.read_csv('Chinas Population En.csv')
print(dataset)
```

Output 10(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile.py"
year total population (ten thousand people) ... Urban population (ten thousand people) Rural population (ten thousand peo
ple)
0 2021 141260 ... 91425 4
9835
1 2020 141212 ... 90220 5
0992
2 2019 141008 ... 88426 5
2582
3 2018 140541 ... 86433 5
4108
4 2017 140011 ... 84343 5
5668
.. ... ...
...
68 1953 58796 ... 7826 5
0970
69 1952 57482 ... 7163 5
0319
70 1951 56300 ... 6632 4
9668
71 1950 55196 ... 6169 4
9027
72 1949 54167 ... 5765 4
8402

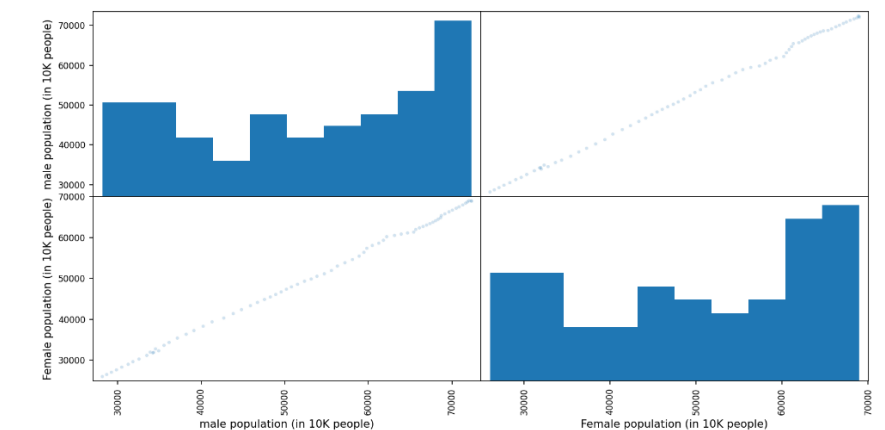
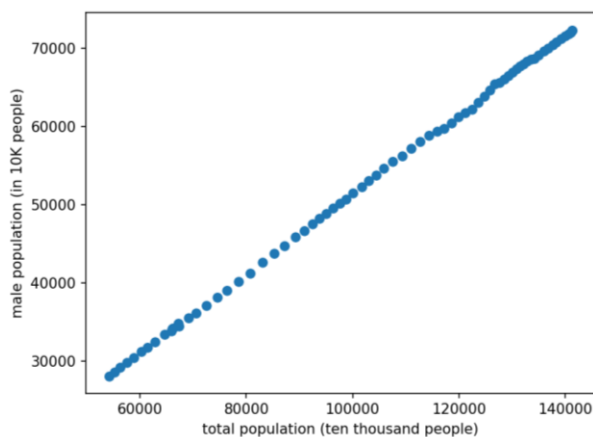
[73 rows x 6 columns]
```

Lab 10(b)

```
plt.scatter(data = dataset, x = 'total population (ten thousand people)' , y = 'male population (in 10K people)' )
plt.xlabel('total population (ten thousand people)')
plt.ylabel('male population (in 10K people)')
plt.show()
```

```
x=['male population (in 10K people)']
y=['Female population (in 10K people)']
z=['male population (in 10K people)','Female population (in 10K people)']
pd.plotting.scatter_matrix(dataset[z], alpha=0.2,
figsize=(10,8))
plt.show()
```

Output 10(b)



Lab 10 (c)

```
import scipy as scipy
from scipy import stats

data = [ 160,120,175,180,182]
hyp_mean = 163

tstatic , pvalue = stats.ttest_1samp(data,hyp_mean)
print(tstatic)
print(pvalue)

#interpreataion
alpha = 0.05
if pvalue < alpha :
    print("Reject hyp_mean, pop mean not equals to
hyp_mean")
else :
    print("fail to reject hyp_mean, pop mean equals to
hyp_mean")
```

Output 10(c)

```
PS E:\4TH SEM\python 4th sem\py exam> python -u "e:\4TH SEM\python 4th sem\py exam\10c.py"
0.03474191421756055
0.9739501143727055
fail to reject hyp_mean, pop mean equals to hyp_mean
PS E:\4TH SEM\python 4th sem\py exam>
```

Lab 10(d)

```
import scipy as scipy
from scipy import stats

data1 = [160,120,175,180,182]
data2 = [10,20,30,40]

tstat, pvalue = stats.ttest_ind(data1,data2)
print(tstat, "\n" ,pvalue)

#interpretation
alpha = 0.05
if pvalue < alpha :
    print("Reject hyp_mean, pop mean not equals to
hyp_mean")
else :
    print("fail to reject hyp_mean, pop mean equals to
hyp_mean")
```

Output 10(d)

```
PS E:\4TH SEM\python 4th sem\py exam> python -u "e:\4TH SEM\python 4th sem\py exam\10d.py"
9.723908629231692
2.5723003181299273e-05
Reject hyp_mean, pop mean not equals to hyp_mean
PS E:\4TH SEM\python 4th sem\py exam> █
```

Lab 11(a)

```
import matplotlib.pyplot as plt

languages = 'html' , 'python', 'java', 'C++'
popularity = [20, 30, 12 ,23]

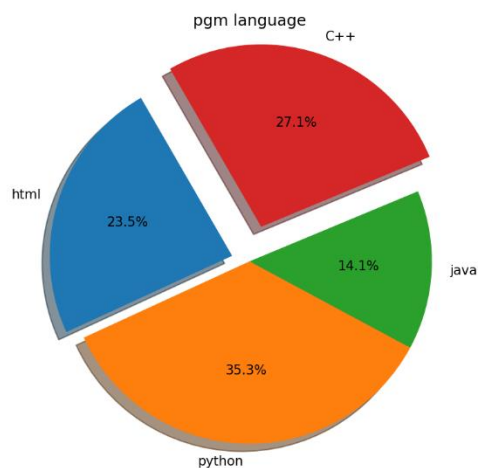
colors = ["#a2345b", "#cc345f", "#er7765", "#er4343"]

explode = (0.1, 0,0,0.2)

plt.pie(popularity,explode=explode, labels = languages,
autopct = '%1.1f%%', shadow = True, startangle=120)

#plt.axis('equal')
plt.title("pgm language")
plt.show()
```

Output 11(a)



Lab 11(b)

```
import numpy as np
import matplotlib.pyplot as plt

groups = 4
men = (4,5,7,2)
women = (8,7,6,5)

fig, ax = plt.subplots()
index = np.arange(groups)

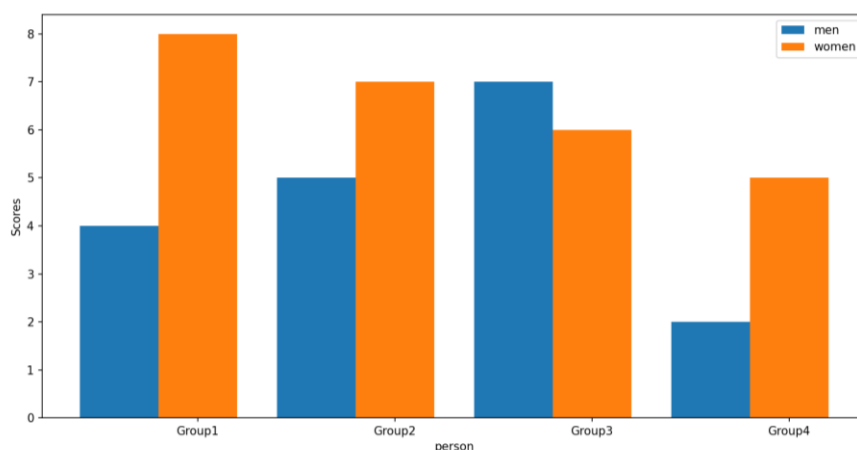
bar_width = 0.40
opacity = 0.75

bar1 = plt.bar(index,men,bar_width, label= 'men' )
bar2 = plt.bar(index + bar_width, women, bar_width , label =
'women')

plt.xlabel('person')
plt.ylabel("Scores")

plt.xticks(index + bar_width, ('Group1', 'Group2', 'Group3',
'Group4'))
plt.legend()
plt.show()
```

Output 11(b)



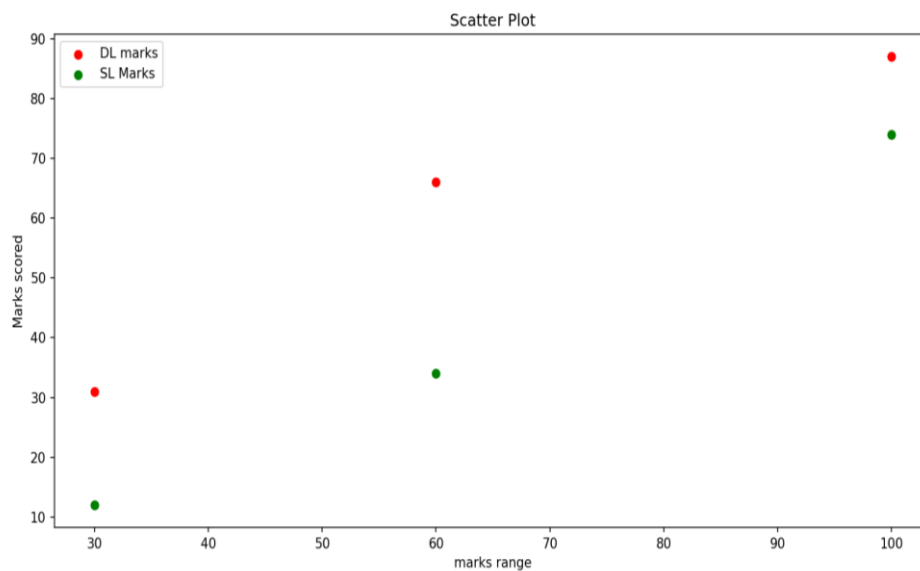
Lab 11(c)

```
import matplotlib.pyplot as plt
import pandas as pd

dlm = [12,34,74]
slm = [31,66,87]
m_range = [30,60,100]

plt.scatter(m_range,slm,label = 'DL marks' , color = 'red')
plt.scatter(m_range,dlm,label = 'SL Marks' , color =
'green')
plt.xlabel('marks range')
plt.ylabel('Marks scored')
plt.title('Scatter Plot')
plt.legend()
plt.show()
```

Output 11(c)



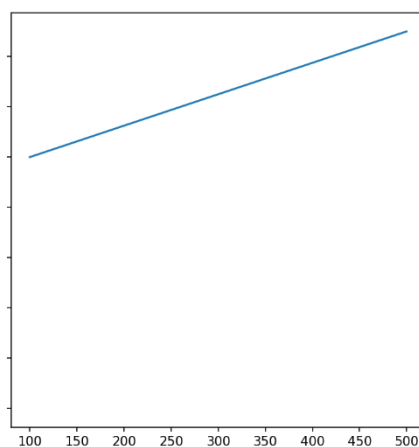
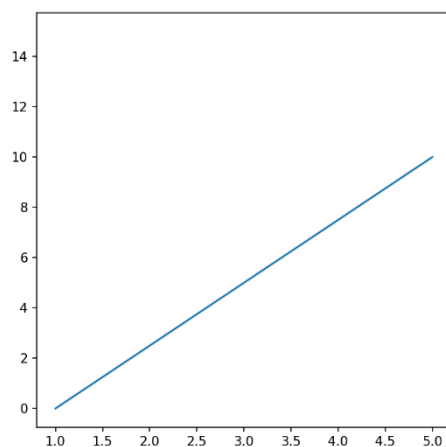
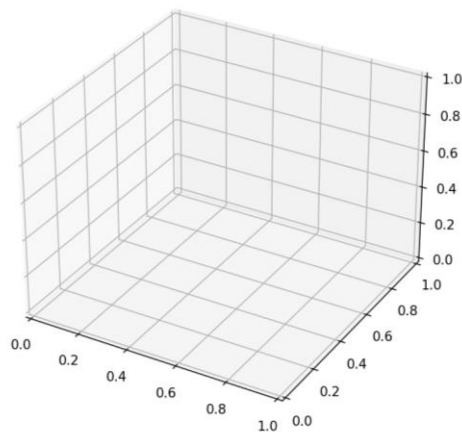
Lab 11(d)

```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
ax = plt.axes(projection = '3d')
fig2, (ax, ax2) = plt.subplots(ncols = 2, sharey = True)
ax.plot([1,5],[0,10])
ax2.plot([100, 500], [10, 15])
ax.set_box_aspect(1)
ax2.set_box_aspect(1)

plt.show()
```

Output 11(d)



Lab 12(a)

```
import numpy as np
data = [5, 8, 12, 15, 18, 20, 22, 23, 25, 28, 30, 32, 35,
40, 42]
mean_manual = sum(data) / len(data)
sorted_data = sorted(data)
if len(sorted_data) % 2 == 0:
    median_manual = (sorted_data[(len(sorted_data)//2)-1] +
sorted_data[(len(sorted_data)//2)]) / 2
else:
    median_manual = sorted_data[(len(sorted_data)//2)]

from collections import Counter
mode_manual = Counter(data).most_common(1)[0][0]
mean = mean_manual
variance = sum([(x - mean) ** 2] for x in data) /
len(data)
standard_deviation_manual = variance ** 0.5
q1_manual = np.percentile(data, 25)
q3_manual = np.percentile(data, 75)
mean_numpy = np.mean(data)
median_numpy = np.median(data)
mode_numpy = np.argmax(np.bincount(data))
standard_deviation_numpy = np.std(data)
q1_numpy = np.percentile(data, 25)
q3_numpy = np.percentile(data, 75)
print("Mean (Manual):", mean_manual)
print("Median (Manual):", median_manual)
print("Mode (Manual):", mode_manual)
print("Standard Deviation (Manual):",
standard_deviation_manual)
print("Q1 (Manual):", q1_manual)
print("Q3 (Manual):", q3_manual)

print("Mean (NumPy):", mean_numpy)
print("Median (NumPy):", median_numpy)
print("Mode (NumPy):", mode_numpy)
```

```
print("Standard Deviation (NumPy):",  
      standard_deviation_numpy)  
print("Q1 (NumPy):", q1_numpy)  
print("Q3 (NumPy):", q3_numpy)
```

Output 12(a)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\12a.py"  
Mean (Manual): 23.666666666666668  
Median (Manual): 23  
Mode (Manual): 5  
Standard Deviation (Manual): 10.693715703886818  
Q1 (Manual): 16.5  
Q3 (Manual): 31.0  
Mean (NumPy): 23.666666666666668  
Median (NumPy): 23.0  
Mode (NumPy): 5  
Standard Deviation (NumPy): 10.693715703886818  
Q1 (NumPy): 16.5  
Q3 (NumPy): 31.0  
PS E:\4TH SEM\SK Python>
```


Lab 12(b)

```
def fact(n):
    res = 1
    for i in range(2, n + 1):
        res = res * i
    return res

def C(n, r):
    return fact(n) / (fact(r) * fact(n - r))

def findWays(arr, N, K):
    arr.sort(reverse=True)
    p = 0
    q = 0

    for i in range(K):
        if (arr[i] == arr[K - 1]):
            p += 1

    for i in range(N):
        if (arr[i] == arr[K - 1]):
            q += 1

    ans = C(q, p)

    return int(ans)

arr = [2, 3, 4, 5, 2, 2]
N = len(arr)
K = 4

print("No. of methods to choose ", K, "elements from", arr,
      "with maxximum sum is : ", findWays(arr, N, K))
```

Output 12(b)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\12b new.py"
No. of methods to choose 4 elements from [5, 4, 3, 2, 2, 2] with maxximum sum is : 3
PS E:\4TH SEM\SK Python> █
```

Lab 12(c)

```
import numpy as np
arr = np.array([[3, 1, 4, 2],
               [8, 6, 5, 7],
               [12, 10, 9, 11]])
sorted_arr = np.sort(arr, axis=1)
print("Original 2D array:")
print(arr)
print("Sorted 2D array (row-wise):")
print(sorted_arr)
```

Output 12(c)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for you. If you want to re-enable it, run 'Import-Module PSReadLine'.

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\12c.py"
```

```
Original 2D array:
```

```
[[ 3  1  4  2]
 [ 8  6  5  7]
 [12 10  9 11]]
```

```
Sorted 2D array (row-wise):
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

```
PS E:\4TH SEM\SK Python>
```

Lab 12(d)

```
import numpy as np
Sample_array = np.array([[21, 21, 5, 10], [51, 32, 83, 24],
[11, 33, 66, 37]])
```

```
Index = 0
```

```
Array_sort = Sample_array[Sample_array[:,Index].argsort()]
```

```
print("The original array is:", "\n", "\n", Sample_array,
"\n")
print("The sorted array is:", "\n", "\n", Array_sort)
```

Output 12(d)

```
PS E:\4TH SEM\SK Python> python -u "e:\4TH SEM\SK Python\tempCodeRunnerFile
The original array is:

[[21 21  5 10]
 [51 32 83 24]
 [11 33 66 37]]

The sorted array is:

[[11 33 66 37]
 [21 21  5 10]
 [51 32 83 24]]
PS E:\4TH SEM\SK Python> █
```