KMM COLLEGE OF ARTS AND SCIENCE

THRIKKAKARA, COCHIN – 21

RECORD BOOK

KMM COLLEGE OF ARTS AND SCIENCE

THRIKKAKARA, COCHIN – 21

RECORD BOOK

Name : STEVE RODRIGUES

Roll No : 16 **Reg. No. : 213242210432**

Subject : PYTHON PROGRAMMING FOR DATA SCIENCE

Course : MCA

Semester : 3 **Year: 2021-2023**

CERTIFICATE

Certified that this is a bonafide record of the practical work done by Mr. **STEVE RODRIGUES** of course **MCA** during academic year 2021-23 for the subject **PYTHON PROGRAMMING FOR DATA SCIENCE** in the college.

Staff in Charge

Name:

Date:

Head of the Department

Name:

Date:

Internal Examiner

External Examiner

INDEX

SL. No	DATE	PROGRAM	Pg. No
		SIMPLE PROGRAMS	
1	04-12-2022	Arithmetic operations	1
2	14-12-2022	Addition of two complex numbers	2
3	14-12-2022	program to read 5 numbers and perform operations-largest and smallest number,absolute value of first number and square root of second number	3
4	14-12-2022	Number is positive,negative or zero	4
5	14-12-2022	Calcuclate total marks,average and print grade	5
6	14-12-2022	Calculator	7
7	14-12-2022	Armstrong or not	9
8	19-12-2022	Palindrome or not	10
9	19-12-2022	Palindrome up to n	11
10	19-12-2022	Armstrong up to n	12
11	19-12-2022	Fibonacci series	13
12	04-01-2023	First n natural numbers	14
13	04-01-2023	Print the series	15
14	04-01-2023	Print the series	16
15	07-01-2023	String operations	17
16	07-01-2023	Maximum and minimum list of numbers	20
17	07-01-2023	Sum of list of numbers	21
18	07-01-2023	Read a string and print all the words and number of words	22
19	07-01-2023	Stack using list	23
20	07-01-2023	Print keys of dictionary in upper case	26
21	07-01-2023	Occurrences of each word in a sentence	27
22	07-01-2023	Count of each character in a string	28
23	07-01-2023	Dictionary of odd and even numbers and its count from a given set of values	29
24	07-01-2023	Calculate the count of vowels, consonants, digits, whitespaces and special character from the given sentence using dictionary	30
25	07-01-2023	Set operations	31
26	07-01-2023	Tuple operations	32
27	12-01-2023	To get a string which is n copies of the given string using function	33

28	12-01-2023	Accept roll no name three marks and print and set anyone of the mark as default argument	34
29	12-01-2023	Accept two list and check whether at least they have one common member using function	35
30	12-01-2023	Remove duplicate elements in a function	36
31	12-01-2023	Longest word using function	37
32	18-01-2023	Calculate sum of two numbers using lambda function	38
33	18-01-2023	To create class student with three data members(rollno,name,marks)member function read and display	39
34	18-01-2023	To implement a class named employee. data members empid,name,salary.member function parameterized constructor and display	40
35	25-01-2023	Define a class to represent a bank account .	41
36	30-01-2023	To implement inheritance.	44
37	30-01-2023	Program to searcxh in a given list of fruits(built-in exception)	45
38	30-01-2023	Perform division operation using exception.	46
39	30-01-2023	Program to read and print voters name and age .raise exception if he or she is not eligible to vote.	47
40	30-01-2023	To read salary value and print the same.Raise user-defined exception if the salary is not in between 5000 and 15000.	48
41	30-01-2023	Program to read the student details and print the same.raise user defined exception .	49
42	01-02-2023	To read a file name and print the number of lines in the given file	51
43	01-02-2023	To read a file and print all the file contents in uppercase.	52
44	01-02-2023	Program to input a file name and print all distinct words in the file with its count.	53
45	01-02-2023	Read two file names where file 1 already exist and copy elements of file1 to file 2. use exception if file 1 is not exist.	54
46	03-02-2023	Program to delete items from particular position using exception	55
47	03-02-2023	Program to retrieve details of a particular student from student database.	56
48	03-02-2023	Program to create a table in a database .insert 5 rows and display it.	57
49	03-02-2023	Menu driven program-print all details of employee,details of particular employee,print details of employee whose salary greater than given value	59

50	03-02-2023	Program to delete details of particular employee and print all contents	61
51	03-02-2023	Program to delete, update details of particular employee and print all contents	62
52	03-02-2023	Database connection program	63
53	03-02-2023	Implementation of 1D array	64
54	03-02-2023	Program to perform arithmetic operation on 2D array	65
		DATA SCIENCE PROGRAMS	
55	01-02-2023	Program to demonstrate line chart.	67
56	01-02-2023	Program to demonstrate barchart.	69
57	06-02-2023	Program to demonstrate scatter plot	71
58	06-02-2023	Program to demonstrate pie chart.	72
59	08-02-2023	Program to demonstrate data cleaning	73
60	13-02-2023	Program to demonstrate linear regression (1D array)	80
61	13-02-2023	Program to create a model and use the created model for prediction	81
62	13-02-2023	Program to create a model and use the model for prediction (CSV)	84
63	15-02-2023	Prediction using logistic regression.	87
64	15-02-2023	Classification	88
65	16-03-2023	clustering	90
		DJANGO PROGRAMS	
66	16-03-2023	Program to create two views	93
67	16-03-2023	Program to create static web page.	96

Program No: 1 **Date: 14/12/2022**

AIM: Write a program to perform all arithmetic operations

SOURCE CODE

```
n=int(input('Enter first number: '))
m=int(input('Enter second number: '))
s=n+m
sub=n-m
mul=n*m
div=n/m
mod=n%m
exp=n**m
fdiv=n//m
print(n,'+',m,'=',s)
print(n,'-',m,'=',sub)
print(n,'*',m,'=',mul)
print(n,'/',m,'=',div)
print(n,'%',m,'=',mod)
print(n,'to the power',n,'=',exp)
print(n,'//',n,'=',fdiv)
```

OUTPUT

```
Enter first number: 10
Enter second number: 2
10 + 2 = 12
10 - 2 = 8
10 * 2 = 20
10 / 2 = 5.0
10 % 2 = 0
10 to the power 10 = 100
10 // 10 = 5
```

Program No: 2 **Date: 14/12/2022**

AIM: Write a program to do addition of two complex numbers

SOURCE CODE

```
a=complex(input('enter First number: '))
b=complex(input('enter Second number: '))
s=a+b
p=a*b;
print('sum is: ',s)
print('product is: ',p)
```

OUTPUT

```
enter First number: 10+3j
enter Second number: 2+4j
sum is:  (12+7j)
product is:  (8+46j)
```

Program No: 3 **Date: 14/12/2022**

AIM: Program to read 5 numbers and perform the following operation

1. Print the largest number
2. Print the smallest number
3. Print absolute value of first number
4. Print the square root of second number

SOURCE CODE

```
import math
a=int(input('Enter the first number: '))
b=int(input('Enter the second number: '))
c=int(input('Enter the third number: '))
d=int(input('Enter the fourth number: '))
e=int(input('Enter the fifth number: '))
p=max(a,b,c,d,e)
q=min(a,b,c,d,e)
print('Maximum number : ',p)
print('Minimum number : ',q)
print('Absolute value',a,'is:',abs(a))
print('Square root of',b,'is:',math.sqrt(b))
```

OUTPUT

```
Enter the first number: -1
Enter the second number: 9
Enter the third number: 8
Enter the fourth number: 5
Enter the fifth number: 4
Maximum number :  9
Minimum number :  -1
Absolute value -1 is: 1
Square root of 9 is: 3.0
```

Program No: 4 **Date: 14/12/2022**

AIM: Program to check whether a number is positive negative or zero

SOURCE CODE

```
n=int(input('Enter a number: '))
if(n>0):
    print('the number is positive')
elif(n==0):
   print('the number is zero')
else:
   print('the number is negetive')
```

OUTPUT

```
Enter a number: -1
the number is negetive
'
```

Program No: 5 **Date: 14/12/2022**

AIM: Write a program to input three marks and calculate total and average then print grade

Average>=90 Grade A
Average>=75 and <90 Grade B
Average>=60 and <75 Grade C
Average>=50and <60 Grade D
Otherwise E Grade

SOURCE CODE

```
m1=int(input('Enter Mark1: '))

m2=int(input('Enter Mark2: '))

m3=int(input('Enter Mark3: '))

total=m1+m2+m3

avg=total/3

print('total marks= ',total)

print('Average mark= ',avg)

if(avg>=90):

   print('Grade=A')

elif(avg>=75 and avg<90):

    print('Grade=B')

elif(avg>=60 and avg<75):

   print('Grade=C')

elif(avg>=50 and avg<60):

   print('Grade=D')

else:

    print('Grade=E')
```

OUTPUT

```
Enter Mark1: 90
Enter Mark2: 98
Enter Mark3: 99
total marks=  287
Average mark=  95.66666666666667
Grade=A
```

Program No: 6 **Date: 14/12/2022**

AIM: write a program to create a calculator (+,-,/,*,//)

SOURCE CODE

```
import math
a=int(input('Enter the first number: '))
b=int(input('Enter the second number: '))
c=input('Enter the operator: ')
if(c=='+'):
    print(a,'+',b,'=',a+b)
elif(c=='-'):
    print(a,'-',b,'=',a-b)
elif(c=='*'):
    print(a,'*',b,'=',a*b)
elif(c=='/'):
    print(a,'/',b,'=',a/b)
elif(c=='//'):
    print(a,'//',b,'=',a//b)
else:
    print('wrong operator')
```

OUTPUT

```
Enter the first number: 80
Enter the second number: 20
Enter the operator: -
80 - 20 = 60
```

```
Enter the first number: 55
Enter the second number: 5
Enter the operator: *
55 * 5 = 275

Enter the first number: 10
Enter the second number: 2
Enter the operator: */
Inavlid Operator!!
```

Program No: 7 **Date: 14/12/2022**

AIM: Program to check the given number is Armstrong or not

SOURCE CODE

```
num = int(input("Enter a number: "))
sum = 0
temp = num
while(temp > 0):
   r= temp % 10
   sum = sum+(r*r*r)
   temp=(int)(temp/10)
ifnum == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")
```

OUTPUT

```
Enter a number: 153
153 is an Armstrong number
```

Program No: 8 **Date: 19/12/2022**

AIM: Program to check a given number is palindrome or not

SOURCE CODE

```
n=int(input("Enter number: "))
temp=n
rev=0
while(n>0):
    num=n%10
    rev=rev*10+num
    n=(int)(n/10)
iftemp==rev:
    print("The number is a palindrome")
else:
    print("The number is not a palindrome")
```

OUTPUT

```
Enter number: 151
The number is a palindrome
```

Program No: 9 **Date: 19/12/2022**

AIM: program to print palindrome numbers up to n

SOURCE CODE

```
m= int(input('Please Enter the limit: '))
print('Palindrome Numbers are: ')
for num in range(1, m + 1):
   temp = num
   rev = 0
   while(temp > 0):
      rem = temp % 10
      rev = (rev * 10) + rem
      temp = (int)(temp/10)
   if num == rev :
      print("%d " %num, end=' ')
```

OUTPUT

```
Please Enter the limit: 100
Palindrome Numbers are:
1  2  3  4  5  6  7  8  9  11  22  33  44  55  66  77  88  99
```

Program No: 10 **Date: 19/12/2022**

AIM: program to print Armstrong numbers up to n

SOURCE CODE

```
m= int(input(' Enter the limit: '))
print('armstrong Numbers are')
for num in range(1,m+1):
   sum = 0
   temp = num
   while(temp > 0):
      r= temp % 10
      sum = (r*r*r)+sum
      temp=(int)(temp/10)
   if(num == sum):
       print("%d " %num, end=' ')
```

OUTPUT

```
 Enter the limit: 500
armstrong Numbers are
0  1  153  370  371  407
'
```

Program No: 11 **Date: 19/12/2022**

AIM: Program to print the Fibonacci series

SOURCE CODE

```
n=int(input("Enter the limit: "))
a=0
b=1
sum=0
count=1
print('Fibonacci Series: ', end = ' ')
while(count <=n):
  print(sum, end = ' ')
  count += 1
  a=b
  b=sum
  sum = a + b
```

OUTPUT

```
Enter the limit: 10
Fibonacci Series:  0 1 1 2 3 5 8 13 21 34
```

Program No: 12 **Date: 04/01/2023**

AIM: Program to print the square root of first n natural numbers using for loop

SOURCE CODE

```
n= int(input('Enter limit: '))
import math
for i in range(1,n):
    s=math.sqrt(i)
    print('square root of ',i,'is: ',s)
```

OUTPUT

```
Enter limit: 10
square root of  1 is:  1.0
square root of  2 is:  1.4142135623730951
square root of  3 is:  1.7320508075688772
square root of  4 is:  2.0
square root of  5 is:  2.23606797749979
square root of  6 is:  2.449489742783178
square root of  7 is:  2.6457513110645907
square root of  8 is:  2.8284271247461903
square root of  9 is:  3.0
```

Program No: 13 **Date: 04/01/2023**

AIM: Print the series

1

1 2

1 2 3

1 2 3 4

SOURCE CODE

```
n= int(input('Enter limit: '))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j,end=' ')
    print()
```

OUTPUT

```
Enter limit: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Program No: 14 **Date: 04/01/2023**

AIM: Print the series

*

* *

* * *

* * * *

* * * * *

SOURCE CODE

```
n= int(input('Enter limit: '))
for i in range(1,n+1):
    for j in range(1,i+1):
        print('*',end=' ')
    print()
else:
    print('end of the program')
```

OUTPUT

```
Enter limit: 5
*
* *
* * *
* * * *
* * * * *
end of the program
```

Program No: 15 **Date: 04/01/2023**

AIM: Write a menu driven program for the following string operations

1. Print the string in upper case
2. Length of a string
3. Replace the occurrence of a Sub string
4. count a Sub String
5. Find whether a substring is present
6. Print all the words in a string
7. Print the given String n times

SOURCE CODE

```
s=input('enter a string: ')
print('1.Upper case')
print('2.Length')
print('3.replace the occurrence of a Sub string')
print('4.count a Sub String')
print('5.Find whether a substring is present')
print('6.Print all the words in a string')
print('7.Print the given String n times')
ch=int(input('enter your choice: '))
if(ch==1):
   print('Upper case=',s.upper())
elif(ch==2):
   print('Length of the string',len(s))
elif(ch==3):
   s1=input('enter the substring: ')
   print('Replacing occurances of',s1,'is: ',s.replace(s1,'x'))
elif(ch==4):
```

```
    s2=input('enter the substring: ')
    print('Count of a substring',s2,'is: ',s.count(s2,0,len(s)))
elif(ch==5):
    s3=input('Enter the substring: ')
    w=s.find(s3)
    if(w==-1):
        print('The givenword not present')
    else:
        print('The word is present')
elif(ch==6):
    print('The words in the given String is: ',s.split())
elif(ch==7):
    n=int(input('Enter the no of times a string want to repeat: '))
    print('String printing ', n ,'times...',s*n)
```

OUTPUT

```
enter a string: Welcome to python
1.Upper case
2.Length
3.replace the occurrence of a Sub string
4.count a Sub String
5.Find whether a substring is present
6.Print all the words in a string
7.Print the given String n times
enter your choice: 1
Upper case= WELCOME TO PYTHON
```

```
enter a string: Welcome to python
1.Upper case
2.Length
3.replace the occurrence of a Sub strir
4.count a Sub String
5.Find whether a substring is present
6.Print all the words in a string
7.Print the given String n times
enter your choice: 5
Enter the substring: Welcome
The word is present
```

Program No: 16 **Date: 04/01/2023**

AIM: Write a program to print the maximum and minimum of a list of numbers

SOURCE CODE

```
s=input('enter a string: ')

l=list(map(int,s.split()))

print(l)

print('length of list: ',len(l))

print('maximum of list: ',max(l))

print('minimum of list: ',min(l))
```

OUTPUT

```
enter a string: 2 4 6 8 10
[2, 4, 6, 8, 10]
length of list:  5
maximum of list:  10
minimum of list:  2
```

Program No: 17 **Date: 07/01/2023**

AIM: Write a program to find the sum of list of numbers

SOURCE CODE

```
s=input('enter a string: ')
l=list(map(int,s.split()))
print(l)
sum=0
for i in l:
    sum=sum+i
print('sum of list: ',sum)
```

OUTPUT

```
enter a string: 1 2 3
[1, 2, 3]
sum of list:  6
```

Program No: 18 **Date: 07/01/2023**

AIM: Write a program to read string and print all the words in the string also print the number of words

SOURCE CODE

```
s=input('Enter a string: ')

print(s)

l=s.split()

print('Number of words in the String: ',len(l))
```

OUTPUT

```
Enter a string: Welcome to python
Welcome to python
Number of words in the String:  3
```

Program No: 19 **Date: 07/01/2023**

AIM: Write a menu driven program to implement stack using list

1. Insert
2. Delete
3. Count
4. Exit

SOURCE CODE

```
s=input('enter the elements in stack: ')
l=list(map(int,s.split()))
print(l)
while(1):
   print('STACK OPERATION')
   print('1.INSERT')
   print('2.DELETE')
   print('3.COUNT')
   print('4.EXIT')
   ch=int(input('Enter your choice: '))
   if ch==1:
      m=int(input('Enter the element to be insertded: '))
      l.append(m)
      print(l)
   elif ch==2:
      if(l==list()):
         print('Empty list')
      else:
```

```
        p=l.pop()
        print('The item deleted is:',p)
    elif ch==3:
        print('The count is:',len(l))
    else:
        exit()
```

OUTPUT

```
Enter the elements in stack: 1 2 3
[1, 2, 3]
STACK OPERATION
1.INSERT
2.DELETE
3.COUNT
4.EXIT
Enter your choice: 1
Enter the element to be insertded: 4
[1, 2, 3, 4]
STACK OPERATION
1.INSERT
2.DELETE
3.COUNT
4.EXIT
Enter your choice: 2
The item deleted is: 4
STACK OPERATION
1.INSERT
2.DELETE
3.COUNT
4.EXIT
```

```
Enter your choice: 3
The count is: 3
STACK OPERATION
1.INSERT
2.DELETE
3.COUNT
4.EXIT
Enter your choice: 4
```

Program No: 20 **Date: 07/01/2023**

AIM: Program to define a dictionary and print all the keys of that dictionary in uppercase

SOURCE CODE

```
d1={'name':'hina','place':'ekm'}
print(d1)
v=d1.keys()
print(v)
for i in v:
    print(i.upper())
```

OUTPUT

```
{'name': 'hina', 'place': 'ekm'}
dict_keys(['name', 'place'])
NAME
PLACE
```

Program No: 21 **Date: 07/01/2023**

AIM: Program to find the occurrences of each word in a sentence

SOURCE CODE

```
s=input('Enter a string: ')
print(s)
l=list(map(str,s.split()))
print(l)
d={}
for i in l:
    if i in d:
        d[i]=d[i]+1
    else:
        d[i]=1
print(d)
```

OUTPUT

```
Enter a string: he is waiting and she is not coming
he is waiting and she is not coming
['he', 'is', 'waiting', 'and', 'she', 'is', 'not', 'coming']
{'he': 1, 'is': 2, 'waiting': 1, 'and': 1, 'she': 1, 'not': 1, 'coming': 1}
```

Program No: 22 **Date: 07/01/2023**

AIM: Program to find the count of each character in a string

SOURCE CODE

```
s=input('Enter a string: ')
print(s)
d={}
for i in s:
    if i in d:
        d[i]=d[i]+1
    else:
        d[i]=1
print(d)
```

OUTPUT

```
Enter a string: see you
see you
{'s': 1, 'e': 2, ' ': 1, 'y': 1, 'o': 1, 'u': 1}
```

Program No: 23 **Date: 07/01/2023**

AIM: Program to create a dictionary of odd and even numbers and its count from a given set of values

SOURCE CODE

```
s=input('Enter the values: ')
print(s)
l=list(map(int,s.split()))
print(l)
dict={'even':0,'odd':0}
for i in l:
    if(i%2==0):
        dict['even']=dict['even']+1
    else:
        dict['odd']=dict['odd']+1
print(dict)
```

OUTPUT

```
Enter the values: 2 4 6 8 1 3 5
2 4 6 8 1 3 5
[2, 4, 6, 8, 1, 3, 5]
{'even': 4, 'odd': 3}
```

Program No: 24 **Date: 07/01/2023**

AIM: Program calculate the count of vowel, consonants, digits, white spaces and special characters from the given sentence using dictionary

SOURCE CODE

```
s=input('Enter a string: ')
print(s)
d={'vowels':0,'cosonants':0,'digits':0,'white_spaces':0,'special_charccter':0}
for i in s:
    if i in 'aeiouAEIOU':
        d['vowels']=d['vowels']+1
    elif i in'bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ':
        d['cosonants']= d['cosonants']+1
    elif i in '0123456789':
        d['digits']=d['digits']+1
    elif i in ' ':
        d['white_spaces']=d['white_spaces']+1
    else:
        d['special_charccter']=d['special_charccter']+1
print(d)
```

OUTPUT

```
Enter a string: Hina@2001 21
Hina@2001 21
{'vowels': 2, 'cosonants': 2, 'digits': 6, 'white spaces': 1, 'special charccter': 1}
```

Program No: 25 **Date: 07/01/2023**

AIM: Program to demonstrate all set operations

SOURCE CODE

```
s={2,4,6,8,10}
s2={1,3,5,7,9}
print('Set operations')
print('length of the set1:',len(s))
print('length of the set1:',len(s2))
print('maximum number in the set1:',max(s))
print('maximum number in the set2:',max(s2))
print('minimum number in the set1:',min(s))
print('minimum number in the set1:',min(s2))
print('sum of the set1:',sum(s))
print('sum of the set2:',sum(s2))
print('union of two sets: ',s.union(s2))
print('Intersection of two sets: ',s.intersection(s2))
print('difference between two sets: ',s.difference(s2))
```

OUTPUT

```
Set operations
length of the set1: 5
length of the set1: 5
maximum number in the set1: 10
maximum number in the set2: 9
minimum number in the set1: 2
minimum number in the set1: 1
sum of the set1: 30
sum of the set2: 25
union of two sets:  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Intersection of two sets:  set()
difference between two sets:  {2, 4, 6, 8, 10}
```

Program No: 26 **Date: 07/01/2023**

AIM: Program to demonstrate all tuple operations

SOURCE CODE

```
t={2,4,6,8,10}

print('Tuple operations')

print('length of the tuple:',len(t))

print('maximum number in the tuple:',max(t))

print('minimum number in the tuple:',min(t))
```

OUTPUT

```
Tuple operations
length of the tuple: 5
maximum number in the tuple: 10
minimum number in the tuple: 2
```

Program No: 27 **Date: 12/01/2023**

AIM: Program to get a string which is n copies of the given string using function

SOURCE CODE

```
def fun(s,n):
    print(s*n)
s=input('enter a string: ')
n=int(input('enter the number of copies: '))
fun(s,n)
```

OUTPUT

```
enter a string: Python
enter how many times a string need to print: 4
PythonPythonPythonPython
```

Program No: 28 **Date: 12/01/2023**

AIM: Program that accepts roll number, name, three marks and print all these. Set any one of the mark as default argument.

SOURCE CODE

```
def details(rno,name,m1,m2,m3=70):
    print('Roll no: ',rno)
    print('Name: ',name)
    print('Mark1: ',m1)
    print('Mark2: ',m2)
    print('Mark3: ',m3)
rno=int(input('enter the roll no: '))
name=input('enter the name: ')
m1=int(input('enter the mark1: '))
m2=int(input('enter the mark2: '))
details(rno,name,m1,m2)
```

OUTPUT

```
enter the roll no: 1
enter the name: Hina
enter the mark1: 50
enter the mark2: 60
Roll no:  1
Name:  Hina
Mark1:  50
Mark2:  60
Mark3:  70
```

Program No: 29 **Date: 12/01/2023**

AIM: Program to accept two list and check whether at least they have one common member using function

SOURCE CODE

```
def lists(l1,l2):
    v=0
    for i in l1:
        for j in l2:
            if i==j:
                v=v+1
                return v
                break
s1=input('Enter a string: ')
l1=list(map(str,s1.split()))
s2=input('Enter a string: ')
l2=list(map(str,s2.split()))
print(l1)
print(l2)
ret=lists(l1,l2)
if ret==1:
        print('common value present')
else:
    print('no common value present')
```

OUTPUT

```
Enter a string: Hina is a girl
Enter a string: Raju is a boy
['Hina', 'is', 'a', 'girl']
['Raju', 'is', 'a', 'boy']
common value present
```

Program No: 30 **Date: 12/01/2023**

AIM: Program to remove duplicate elements in a function

SOURCE CODE

```
def remove(l):
    flist = []
    for i in l:
        if i not in flist:
            flist.append(i)
    return flist
s1=input('Enter a string: ')
l= list(map(int,s1.split()))
print(l)
s=remove(l)
print(s)
```

OUTPUT

```
Enter a string: 2 4 6 8 8 10 10
[2, 4, 6, 8, 8, 10, 10]
[2, 4, 6, 8, 10]
```

Program No: 31 **Date: 12/01/2023**

AIM: Program to print the longest word using function

SOURCE CODE

```
def longest(s):
    length=0
    w = ''
    for i in s.split():
        if(len(i) > length):
            length = len(i)
            w = i
    return (w)


s= input("Enter the string: ")
w = longest(s)
print('Longest word is ',w)
```

OUTPUT

```
Enter the string: Welcome to Python Programming
Longest word is  Programming
```

Program No: 32 **Date: 18/01/2023**

AIM: Write a program to calculate sum of two numbers using lambda function

SOURCE CODE

```
sum=lambda x,y:x+y

x=int(input('Enter the first value: '))

y=int(input('Enter the second value: '))

print('sum= ',sum(x,y))
```

OUTPUT

```
Enter the first value: 15
Enter the second value: 25
sum=  40
```

Program No: 33 **Date: 18/01/2023**

AIM: Write a program to create class student with 3 data members (roll no, name and marks) member function read and display

SOURCE CODE

```
class student:
    def read(self,rno,name,m):
        self.rno=rno
        self.name=name
        self.m=m
    def display(self,rno,name,m):
        print(self.rno)
        print(self.name)
        print(self.m)
p=student()
rno=int(input('Enter the rollno: '))
name=input('Enter the name: ')
m=int(input('Enter the mark: '))
p.read(rno,name,m)
p.display(rno,name,m)
```

OUTPUT

```
Enter the rollno: 12
Enter the name: Hina
Enter the mark: 89
12
Hina
89
```

Program No: 34 **Date: 18/01/2023**

AIM: Create class name employee. Data members empid, name, salary. Member functions parameterized constructor and display. Write a program to implement this class.

SOURCE CODE

```
class employee:
    def __init__(self,empid,name,salary):
        self.empid=empid
        self.name=name
        self.salary=salary
    def display(self):
        print('Employee id: ',self.empid)
        print('Name: ',self.name)
        print('salary: ',self.salary)
e=int(input('Enter the employee id: '))
n=input('enter the Name: ')
s=int(input('Enter the employee salary: '))
p=employee(e,n,s)
p.display()
```

OUTPUT

```
Enter the employee id: 101
enter the Name: Edwin
Enter the employee salary: 50000
Employee id:  101
Name:  Edwin
salary:  50000
```

Program No: 35 **Date: 25/01/2023**

AIM: Define a class to represent Bank account include the following details like name of the depositor, account no, type of account, balance amount in the account. Write a method to assign initial values, to deposit, withdrawal and checking balance.

SOURCE CODE

```
class bank:
    def __init__(self,name,actype,acno):
        self.name=name
        self.acno=acno
        self.actype=actype
        self.balance=500
    def deposite(self):
        amt=float(input('Enter the amount to deposite:'))
        self.balance=self.balance+amt
        print('deposite is :',amt)
    def withdraw(self):
        amt=float(input('Enter the amount to withdraw:'))
        if(amt>self.balance):
            print('Insufficient Balance')
        elif(self.balance==amt):
            print('Withdrawal is not possible ')
            print('Minimum balance is 500')
        else:
            self.balance=self.balance-amt
            print('Amount withdrawn is:',amt)
    def display(self):
        print('name:',self.name)
        print('Ac No:',self.acno)
        print('Account Type:',self.actype)
```

```
        print('Current Balance:',self.balance)
name=input('Enter the name: ')
acno=int(input('Enter the Ac No: '))
actype=input('Enter the account type: ')
b=bank(name,actype,acno)
while(1):
    print('1.Deposite')
    print('2.Withdraw')
    print('3.Display')
    print('4.Exit')
    ch=int(input('Enter your choice:'))
    if(ch==1):
        b.deposite()
    elif(ch==2):
        b.withdraw()
    elif(ch==3):
        b.display()
    else:
        exit()
```

OUTPUT

```
Enter the name: Hina
Enter the Ac No: 123660009800678
Enter the account type: Current
1.Deposite
2.Withdraw
3.Display
4.Exit
Enter your choice:1
Enter the amount to deposite:100000
deposite is : 100000.0
1.Deposite
2.Withdraw
3.Display
4.Exit
Enter your choice:2
Enter the amount to withdraw:10000
Amount withdrwen is: 10000.0
1.Deposite
2.Withdraw
3.Display
4.Exit

Enter your choice:3
name: Hina
Ac No: 123660009800678
Account Type: Current
Current Balance: 90500.0
1.Deposite
2.Withdraw
3.Display
4.Exit
Enter your choice:4
```

Program No: 36 **Date: 30/01/2023**

AIM: Create a base class named polygon with attributes number of sides and values of sides and methods – constructor, input sides, display. Derive a class named triangle from polygon with methods constructor and find area

SOURCE CODE

```
class polygon:
    def __init__(self,nsides):
        self.ns=nsides
        self.l=[]
    def inputside(self):
        s=input(" Enter the sides : ")
        self.l=list(map(int,s.split()))
    def disp(self):
        print(" No of sides ",self.ns)
        print(" Values of sides :",self.l)
class triangle(polygon):
    def __init__(self,ns):
        polygon.__init__(self,ns)
    def area(self):
        a,b,c=self.l
        s=(a+b+c)/2
        a=(s*(s-a)*(s-b)*(s-c))**0.5
        print("Area = ",a)
n=int(input(" Enter the no of sides : "))
t=triangle(n)
t.inputside()
t.disp()
t.area()
```

OUTPUT

```
 Enter the no of sides : 3
 Enter the sides : 4 6 8
 No of sides  3
 Values of sides : [4, 6, 8]
Area =  11.61895003862225
```

Program No: 37 **Date: 30/01/2023**

AIM: Write a program in python to do search in a given list of fruits (use built in exceptions).

SOURCE CODE

```
s=input("Enter set of fruits : ")
l=list(s.split())
n=int(input("Enter a position :"))
try:
   print(l[n])
except IndexError:
   print("There is an error !!! Index out of range ")
```

OUTPUT

```
Enter set of fruits : apple orange
Enter a position :1
orange

Enter set of fruits : apple orange
Enter a position :2
There is an error!!! Index out of range
```

Program No: 38 **Date: 30/01/2023**

AIM: Write a program in python to perform division operation (use exception-try, except, else).

SOURCE CODE

```
a=int(input('Enter the first number:'))
b=int(input('Enter the second number:'))
try:
    c=a/b
except ZeroDivisionError:
    print('Not Divisible By Zero')
else:
    print('successfully completed')
    print(a,'/',b,'=',c)
```

OUTPUT

```
Enter the first number:10
Enter the second number:2
successfully completed
10 / 2 = 5.0

Enter the first number:10
Enter the second number:0
Not Divisible By Zero
```

Program No: 39 **Date: 30/01/2023**

AIM: Write a program in python to read and print voters name and age. Raise exception if he/she is not eligible to vote.

SOURCE CODE

```
name=input(" Enter the name : ")

age=int(input(" Enter the age : "))

try:

   if(age>=18):

      print(" Eligible for voting ")

      print(" Name = ",name)

      print(" Age = ",age)

   else:

      raise ValueError()

except ValueError:

   print(" Not Eligible for voting ")
```

OUTPUT

```
Enter the name : Hina
Enter the age : 19
Eligible for voting
Name =  Hina
Age =  19


Enter the name : Hari
Enter the age : 15
Not Eligible for voting
```

Program No: 40 **Date: 30/01/2023**

AIM: Write a program to read salary value and print the same. Raise user-defined exception if the salary is not in between 5000 and 15000.

SOURCE CODE

```
class UserError(Exception):
    def __init__(self,msg):
        self.msg=msg
    def pmsg(self):
        return(self.msg)
try:
    sal=int(input('Enter the salary: '))
    if(sal<5000 or sal>15000):
        raise(UserError('salary is not Between 5000 and 15000'))
    else:
        print('salary:',sal)
except UserError as E:
    print(E.pmsg())
```

OUTPUT

```
Enter the salary: 14000
salary: 14000

Enter the salary: 1000
salary is not Between 5000 and 15000
```

Program No: 41 **Date: 30/01/2023**

AIM: Write a program to read the student details (roll no, name, mark) and print the same. Raise user-defined exception if the roll no is negative and if the mark is not in between 0 and 600.

SOURCE CODE

```
rollno=int(input('Enter the roll no:'))

name=input('Enter the Name:')

mark=int(input('Enter the mark:'))

class RollError(Exception):

    def __init__(self,msg):

        self.msg=msg

    def pmsg(self):

        return(self.msg)

try:

    if(rollno<0):

        raise(RollError('Roll must be positive'))

except RollError as E:

    print(E.pmsg())

class MarkError(Exception):

    def __init__(self,msg):

        self.msg=msg

    def pmsg(self):

        return(self.msg)

try:

    if(mark<0 or mark>600):
```

```
        raise(MarkError('Mark is not Between 0 and 600'))
    else:
        print('STUDENT DETAILS')
        print('Roll Number:',rollno)
        print('Name:',name)
        print('Mark:',mark)
except MarkError as E:
    print(E.pmsg())
```

OUTPUT

```
Enter the roll no:1
Enter the Name:Advik
Enter the mark:450
STUDENT DETAILS
Roll Number: 1
Name: Advik
Mark: 450

Enter the roll no:-1
Enter the Name:Hina
Enter the mark:780
Roll must be positive
Mark is not Between 0 and 600
```

Program No: 42 **Date: 01/02/2023**

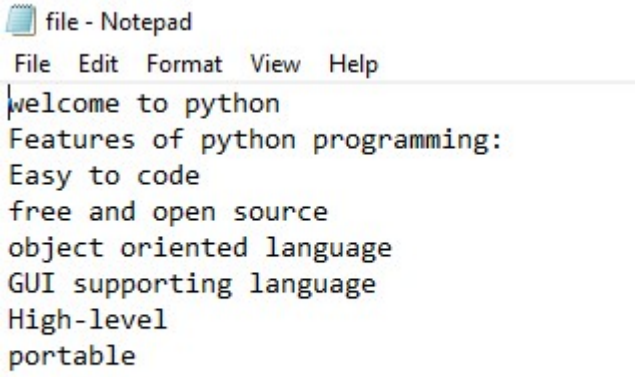
AIM: Write a Python program to read a filename and print the number of lines in the given file.

SOURCE CODE

```
n=input('enter the file Name: ')

f1=open(n,'r')

s=f1.readlines()

print(s)

l=len(s)

print("number of lines",l)

f1.close()
```

OUTPUT

```
enter the file Name: file.txt
['welcome to python\n', 'Features of python programming:\n', 'Easy to code\n', 'free and open sourc
e\n', 'object oriented language\n', 'GUI supporting language\n', 'High-level\n', 'portable\n']
number of lines 8
```



Program No: 43 **Date: 01/02/2023**

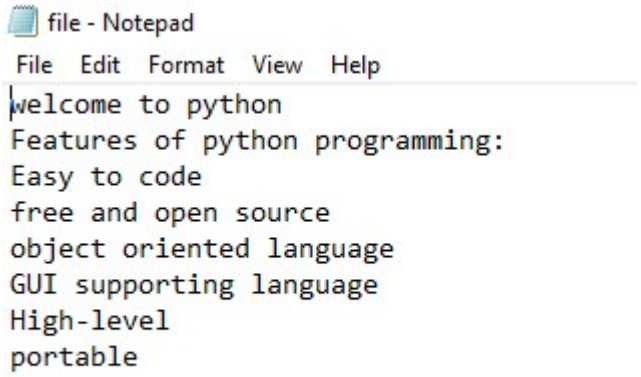
AIM: Write a Python program to read a file and print all the file contents in uppercase.

SOURCE CODE

```
n=input('enter the file Name: ')

f1=open(n,'r')

s=f1.read()

l=s.upper()

print(l)

f1.close()
```

OUTPUT

```
enter the file Name: file.txt
WELCOME TO PYTHON
FEATURES OF PYTHON PROGRAMMING:
EASY TO CODE
FREE AND OPEN SOURCE
OBJECT ORIENTED LANGUAGE
GUI SUPPORTING LANGUAGE
HIGH-LEVEL
PORTABLE
```



Program No: 44 **Date: 01/02/2023**

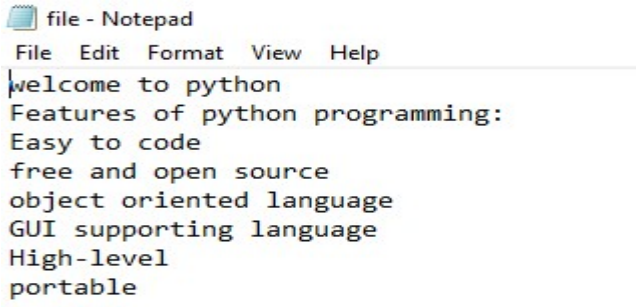
AIM: Write a Python program to input a filename and print all the distinct words in the file with its count

SOURCE CODE

```
n=input("Enter filename:")
f1=open(n,"r")
s=f1.read()
d={}
l=list(s.split())
for i in l:
    if i in d:
        d[i]=d[i]+1
    else:
        d[i]=1
print(d)
f1.close()
```

OUTPUT

```
Enter filename:file.txt
{'welcome': 1, 'to': 2, 'python': 2, 'Features': 1, 'of': 1, 'programming:': 1, 'Easy': 1, 'code':
1, 'free': 1, 'and': 1, 'open': 1, 'source': 1, 'object': 1, 'oriented': 1, 'language': 2, 'GUI': 1
, 'supporting': 1, 'High-level': 1, 'portable': 1}
```



Program No: 45 **Date: 01/02/2023**

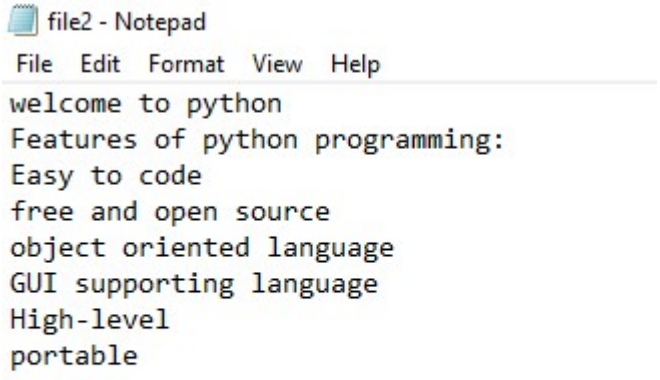
AIM:. Program to read two file names(file1, file2)where file 1 already exist copy the elements of file1 to file 2 use exception if file 1 is not exist

SOURCE CODE

```
try:
    n=input('Enter source file name:')
    f1=open(n,'r')
    s=f1.read()
except FileNotFoundError:
    print('File not found')
d=input('Enter the destination file name:')
f2=open(d,'w')
f2.write(s)
print("Contents are Successfully copied ")
f1.close()
f2.close()
```

OUTPUT

```
Enter source file name:file.txt
Enter the destination file name:file2.txt
Contents are Successfully copied
```



Program No: 46 **Date: 03/02/2023**

AIM: write a python program to delete from a specified person in the given file. Raise an exception if the location of the sentence is not exist.

SOURCE CODE

```
try:
   n=input("Enter file name:")
   f1=open(n,"r")
   s=f1.readlines()
   print(s)
   p=int(input("Enter a position for delete"))
   g=s.pop(p)
   print("item deleted")
   f1.close()
   f2=open(n,"w")
   f2.writelines(s)
   print(s)
   f2.close()
except FileNotFoundError:
   print("value is not in the list")
```

OUTPUT

```
Enter file name:FILE2.TXT
['welcome to python\n', 'Features of python programming:\n', 'Easy to code\n', 'GUI supporting lang
uage\n', 'portable\n']
Enter a position for delete:2
item deleted= Easy to code

FILE ITEMS AFTER DELETING
['welcome to python\n', 'Features of python programming:\n', 'GUI supporting language\n', 'portable
\n']
```

Program No: 47 **Date: 03/02/2023**

AIM: Program to retrieve details of a particular student from student database.

SOURCE CODE

```
import sqlite3

sc=sqlite3.connect('college.db')

c=sc.cursor()

query="select * from student where Roll_no=?"

Roll_no=int(input('Enter the roll number:'))

data=(Roll_no,)

c.execute(query,data)

print('STUDENT DETAILS')

for i in c:

    print(i)

c.close()

sc.close()
```

OUTPUT

```
sqlite> .open college.db
sqlite> select * from student;
101|Wednesday|MCA
102|Enid|MBA
103|Xavier|MCA
104|Tyler|MBA
sqlite>
```

```
Enter the roll number:101
STUDENT DETAILS
(101, 'Wednesday', 'MCA')
```

Program No: 48 **Date: 03/02/2023**

Aim: Write a program to create a table employee in company database with following structure employee id primary key,name,salary.Write a python program to insert 5 rows into employee tables and display all contents.

SOURCE CODE:

```
import sqlite3

s=sqlite3.connect("company.db")

c=s.cursor()

ch='y'

while ch=='y':

    empid=int(input("enter employee id:"))

    name=input("enter employee name:")

    salary=int(input("enter salary:"))

    query="insert into employee1 values(?,?,?)"

    data=(empid,name,salary,)

c.execute(query,data)

ch=input("do you want to continue(y/n):")

s.commit()

for i in c:

    print(i)

query1="select * from employee1"

c.execute(query1)

for i in c:

    print(i)

c.close()

s.close()
```

OUTPUT:

```
enter employee id:1
enter employee name:ameer
enter salary:14000
do you want to continue(y/n):y
enter employee id:2
enter employee name:anu
enter salary:34000
do you want to continue(y/n):y
enter employee id:3
enter employee name:asna
enter salary:15000
do you want to continue(y/n):y
enter employee id:4
enter employee name:ami
enter salary:15000
do you want to continue(y/n):y
enter employee id:5
enter employee name:tom
enter salary:16000
do you want to continue(y/n):n
(1, 'ameer', 14000)
(2, 'anu', 34000)
(3, 'asna', 15000)
(4, 'ami', 15000)
(5, 'tom', 16000)
```

```
sqlite> select * from employee1;
1|ameer|14000
2|anu|34000
3|asna|15000
4|ami|15000
5|tom|16000
```

Program No: 49 **Date: 03/02/2023**

AIM: Write python program to do following

1. print all details of employee
2. print details of particular employee
3. print details of employee whose salary greater than given value

SOURCE CODE

```
import sqlite3
sc=sqlite3.connect('company.db')
c=sc.cursor()
while(1):
   print('1.Display')
   print('2.Details of particular employee')
   print('3.Details of employee whose salary Grater than the given value')
   print('4.Exit')
   ch=int(input('Enter the choice:'))
   if ch==1:
        q="select * from employee"
        c.execute(q)
        for i in c:
            print(i)
   elif ch==2:
        query="select * from employee where empid=?"
        r=int(input("Enter employee id:"))
        data=(r,)
        c.execute(query,data)
        result=c.fetchone()
        if(result==None):
            print("Employee not exist")
        else:
            print(result)
   elif ch==3:
```

```
        query="select * from employee where salary>?"
        salary=input("Enter the salary to check:")
        data=(salary,)
        c.execute(query,data)
        result=c.fetchall()
        print(result)
c.close()
sc.close()
```

OUTPUT

```
1.Display
2.Details of particular employee
3.Details of employee whose salary Grater than the given value
4.Exit
Enter the choice:1
(101, 'Hina', 50000)
(102, 'Hari', 45000)
(103, 'Seb', 35000)
(104, 'Krithika', 45000)
(105, 'Liliya', 80000)
1.Display
2.Details of particular employee
3.Details of employee whose salary Grater than the given value
4.Exit
Enter the choice:2
Enter employee id:104
(104, 'Krithika', 45000)


1.Display
2.Details of particular employee
3.Details of employee whose salary Grater than the given value
4.Exit
Enter the choice:3
Enter the salary to check:45000
[(101, 'Hina', 50000), (105, 'Liliya', 80000)]
1.Display
2.Details of particular employee
3.Details of employee whose salary Grater than the given value
4.Exit
Enter the choice:4
```

Program No: 50 **Date: 03/02/2023**

AIM: Write program to delete details of particular employee and print all content.

SOURCE CODE

```
import sqlite3
sc=sqlite3.connect('company.db')
ch='yes'
c=sc.cursor()
while(ch=='yes'):
    q="delete from employee where empid=?"
    empid=int(input("Enter employee id:"))
    data=(empid,)
    c.execute(q,data)
    print('Item is deleted.....')
    ch=input('Do you want to continue...')
sc.commit()
q="select * from employee"
c.execute(q)
for i in c:
    print(i)
c.close()
sc.close()
```

OUTPUT

```
Enter employee id:103
Item is deleted.....
Do you want to continue...no
(101, 'Hina', 50000)
(102, 'Hari', 45000)
(104, 'Krithika', 45000)
(105, 'Liliya', 80000)
```

Program No: 51 **Date: 03/02/2023**

AIM: Write program to delete update of particular employee and print all content.

SOURCE CODE

```
import sqlite3

sc=sqlite3.connect('company.db')

c=sc.cursor()

query="update  employee  set salary = ? where empid = ?"

salary=int(input ('Enter salary to update:'))

empid=int(input('Enter employee id:'))

data=(salary,empid)

c.execute(query,data)

sc.commit()

q="select * from employee"

c.execute(q)

for i in c:

        print(i)

c.close()

sc.close()
```

OUTPUT

```
Enter salary to update:55000
Enter employee id:101
(102, 'Hari', 45000)
(104, 'Krithika', 45000)
(105, 'Liliya', 80000)
(101, 'Hina', 55000)
```

Program No: 52 **Date: 03/02/2023**

AIM: Create a database name company in sqlite3. Create two tables employee and department in company database.

Structure of department table (dept id primary key, name, location)

Structure of employee table(empid primary key ,name ,location)

Insert 3 rows in department, 5 rows in employee. Write a python program to display all the employee details in a particular department.

SOURCE CODE

```
import sqlite3

sc=sqlite3.connect('company2.db')

c=sc.cursor()

dep=input('Enter department:')

q='select e.empid,e.emp_name,e.emp_loc,d.dept_name from employee e,dept d where d.deptid=e.deptid and d.dept_name=?';

data=(dep,)

c.execute(q,data)

for i in c:

    print(i)
```

OUTPUT

```
Enter department:HR
(103, 'Sebin', 'KLM', 'HR')
```

Program No: 53 **Date: 03/02/2023**

AIM: One dimensional array

SOURCE CODE

```
import numpy as np
n=input('Enter elements:')
arr=list(map(int,n.split()))
print(arr)
p=int(input('Enter position:'))
num=int(input('Enter element to insert:'))
arr=np.insert(arr,p,num)
print(arr)
p=int(input('Enter position to be delete:'))
arr=np.delete(arr,p)
print(arr)
```

OUTPUT

```
Enter elements:2 3 4 5
[2, 3, 4, 5]
Enter position:3
Enter element to insert:6
[2 3 4 6 5]
Enter position to be delete:1
[2 4 6 5]
```

Program No: 54 **Date: 03/02/2023**

AIM: Program to perform arithmetic operation on two dimensional array

SOURCE CODE

```
numpy as np
r1=int(input('enter no of rows of matrix1:'))
c1=int(input('enter no of columns of matrix1:'))
s=input("enter elements of matrix1:")
m1=np.array(list(map(int,s.split())))
m1=m1.reshape(r1,c1)
print("First Matrix:\n",m1)
r2=int(input('enter no of rows of matrix2:'))
c2=int(input('enter no of columns of matrix2:'))
s=input("enter elements of matrix2:")
m2=np.array(list(map(int,s.split())))
m2=m2.reshape(r2,c2)
print("Second  Matrix:\n",m2)
ch=0
print("Matrix Operations\n")
print("1.Addition\n2.Substraction\n3.Multiplication\n4.Division\n5.TRanspose\n6.row
sum,column sum,sum of elements\n7.Exit")
while ch!=7:
   ch=int(input("enter your choice:"))
   if ch==1:
      m3=m1+m2
      print("Addition of m1 and m2:\n",m3)
   elif ch==2:
      m3=m1-m2
      print("Substraction of m1 and m2:\n",m3)
   elif ch==3:
      m3=m1.dot(m2)
      print("multiplication of m1 and m2:\n",m3)
   elif ch==4:
      m3=m1/m2
      print("division of m1 and m2:\n",m3)
   elif ch==5:
      print("transpose of matrix1:\n",m1.transpose())
      print("transpose of matrix2:\n",m2.transpose())
   elif ch==6:
      print("row sum of matrix1:\n",m1.sum(axis=1))
      print("column sum of matrix1:\n",m1.sum(axis=0))
      print("sum of all elements of matrix1:",m1.sum())
      print("row sum of matrix2:\n",m2.sum(axis=1))
      print("column sum of matrix 2:\n",m2.sum(axis=0))
      print("sum of all elements of matrix 2:\n",m2.sum())
```

```
else:
    print("exit...")
    exit
```

OUTPUT

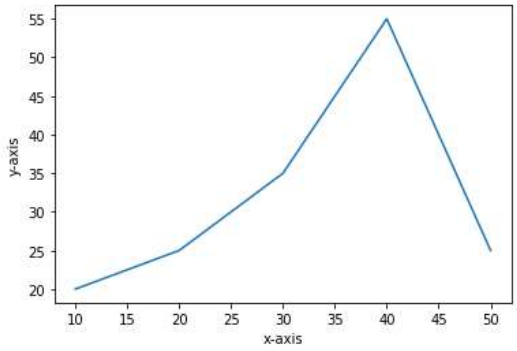
```
enter no of rows of matrix1:2
enter no of columns of matrix1:2
enter elements of matrix1:1 2 3 4
First Matrix:
 [[1 2]
 [3 4]]
enter no of rows of matrix2:2
enter no of columns of matrix2:2
enter elements of matrix2:6 7 8 9
Second  Matrix:
 [[6 7]
 [8 9]]
Matrix Operations

1.Addition
2.Substraction
3.Multiplication
4.Division
5.TRanspose
6.row sum,column sum,sum of elements
7>Exit
enter your choice:1
Addition of m1 and m2:
 [[ 7  9]
 [11 13]]
enter your choice:2
Substraction of m1 and m2:
 [[-5 -5]
 [-5 -5]]
enter your choice:3
multiplication of m1 and m2:
 [[22 25]
 [50 57]]
enter your choice:4
division of m1 and m2:
 [[0.16666667 0.28571429]
 [0.375      0.44444444]]
enter your choice:5
```

DATA SCIENCE

1. Program to demonstrate the line chart for the following data.

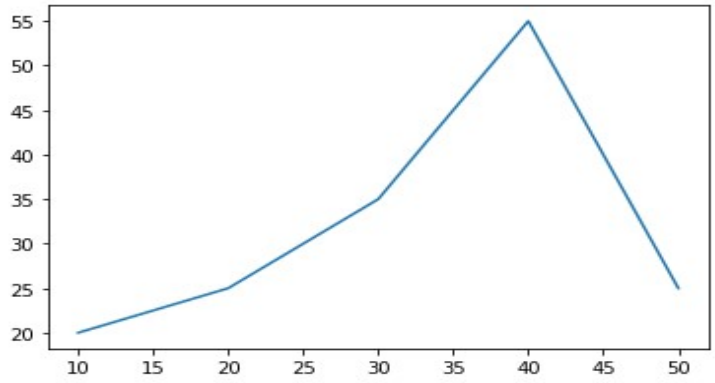
X	Y
10	20
20	25
30	35
40	55
50	25

```
In [7]: x=[10,20,30,40,50]
        y=[20,25,35,55,25]
        plt.plot(x,y)
        plt.ylabel("y-axis")
        plt.xlabel("x-axis")
        plt.show()
```




```
x=[10,20,30,40,50]

y=[20,25,35,55,25]
plt.plot(x,y)
plt.ylabel("y-axis")
plt.xlabel("x-axis")
plt.show()
```

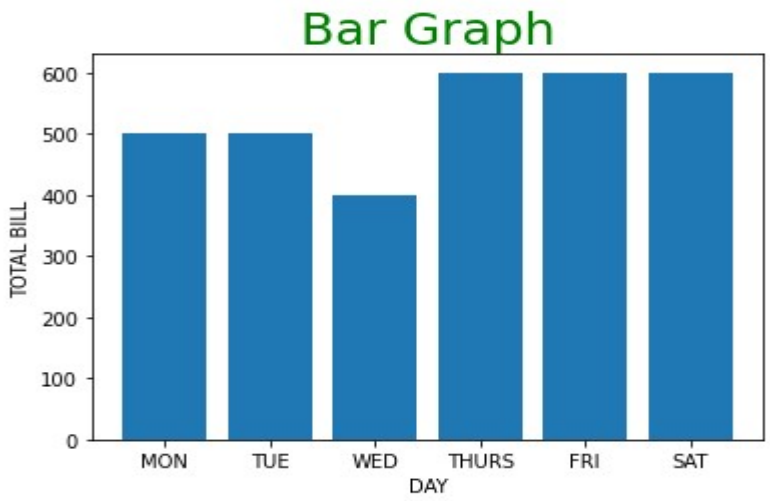
```
In [1]: import matplotlib.pyplot as plt

In [2]: x=[10,20,30,40,50]
        y=[20,25,35,55,25]

In [3]: plt.plot(x,y)
        plt.show()
```



2. Program to demonstrate bar chart for the data

	A	B	C	D
1	DAY	TOTAL BILL		
2	Monday	500		
3	Tuesday	600		
4	Wednesd	70		
5	Thursday	900		
6	Friday	700		
7	Saturday	60		
8	Monday	80		
9	Tuesday	89		
10	Wednesd	12		
11	Thursday	47		
12	Friday	67		
13	Saturday	75		
14	Monday	65		
15	Tuesday	43		
16	Wednesd	44		
17	Thursday	37		
18	Friday	70		
19	Saturday	900		
20	Monday	700		
21	Tuesday	60		
22	Wednesd	80		
23	Thursday	89		
24	Friday	90		
25	Saturday	678		
26				

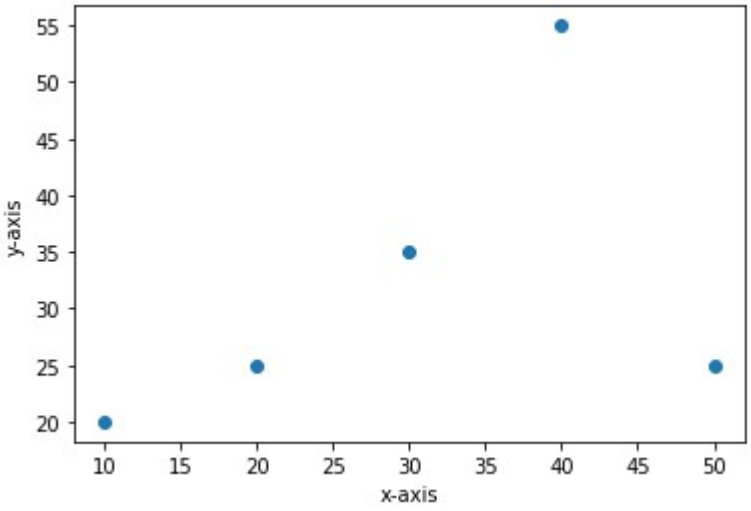
```
df=p.read_csv("barchart.csv")
day=df['DAY']
totalbill=df['TOTAL BILL']
plt.bar(day,totalbill)
plt.title("tips dataset")
plt.title("Bar Graph",fontsize=25,color="green")
plt.ylabel("TOTAL BILL")
plt.xlabel("DAY")
plt.show()
```



	A	B	C
1	DAY	TOTAL BILL	
2	MON	500	
3	TUE	300	
4	WED	400	
5	THURS	100	
6	FRI	200	
7	SAT	400	
8	MON	400	
9	TUE	500	
10	WED	300	
11	THURS	200	
12	FRI	500	
13	SAT	600	
14	MON	200	
15	TUE	500	
16	WED	400	
17	THURS	600	
18	FRI	400	
19	SAT	300	
20	MON	500	
21	TUE	200	

3. Program to demonstrate scatterplot.

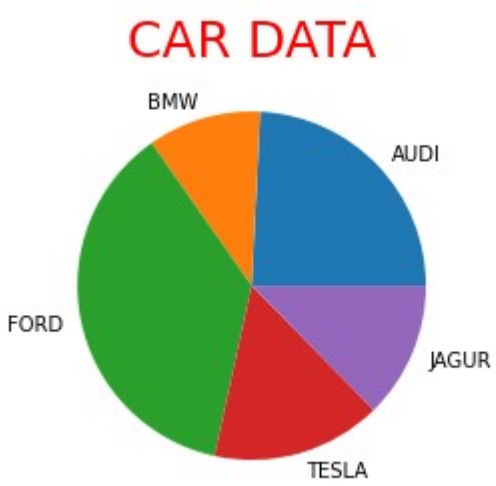
```
import matplotlib.pyplot as plt
x=[10,20,30,40,50]
y=[20,25,35,55,25]
plt.scatter(x,y)
plt.ylabel("y-axis")
plt.xlabel("x-axis")
plt.show()
```



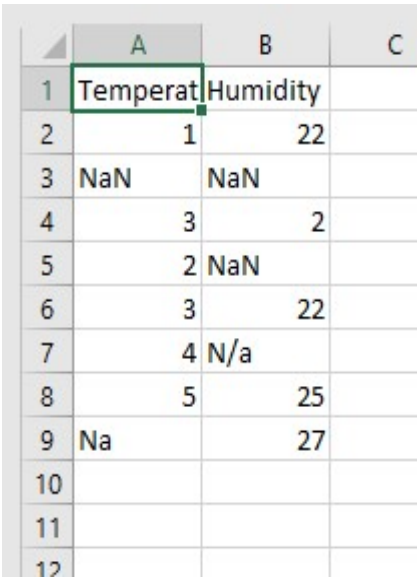
4. Program to demonstrate pie chart for the following data

CARS	SALES %
AUDI	23
BMW	10
FORD	35
TESLA	15
JAGUAR	12

```
import matplotlib.pyplot as plt
cars=['AUDI','BMW','FORD','TESLA','JAGUR']
data=[23,10,35,15,12]
plt.pie(data,labels=cars)
plt.title("CAR DATA",fontsize=25,color="red")
plt.show()
```



5. Sample program to demonstrate data cleaning



	A	B	C
1	Temperat	Humidity	
2	1	22	
3	NaN	NaN	
4	3	2	
5	2	NaN	
6	3	22	
7	4	N/a	
8	5	25	
9	Na	27	
10			
11			
12			

```
import pandas as p
df=p.read_csv('s.csv')
df
```

```
In [18]: import pandas as p
         df=p.read_csv('s.csv')
         df
```

Out[18]:

	Temperature	Humidity
0	1	22
1	NaN	NaN
2	3	2
3	2	NaN
4	3	22
5	4	N/a
6	5	25
7	Na	27

```
df.isnull()
```

Out[19]:

	Temperature	Humidity
0	False	False
1	True	True
2	False	False
3	False	True
4	False	False
5	False	False
6	False	False
7	False	False

df.isnull().sum()

```
In [20]: df.isnull().sum()
Out[20]: Temperature    1
         Humidity       2
         dtype: int64
```

import numpy as np
missing_values=[np.nan,'Na','N/a']
df=p.read_csv('s.csv',na_values=missing_values)

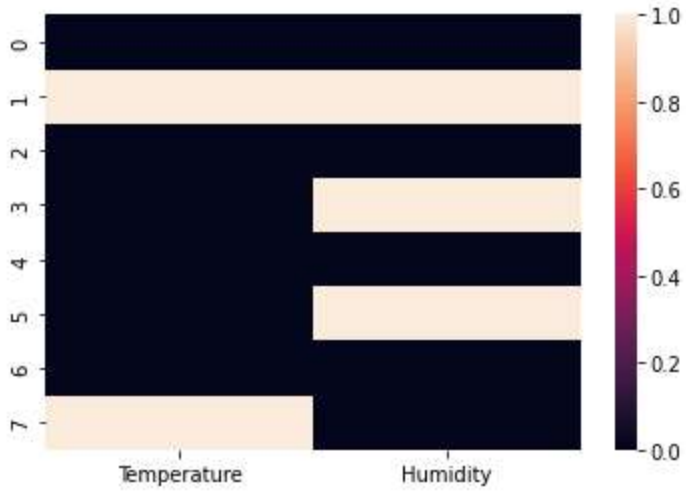
df.isnull()

Out[23]:

	Temperature	Humidity
0	False	False
1	True	True
2	False	False
3	False	True
4	False	False
5	False	True
6	False	False
7	True	False

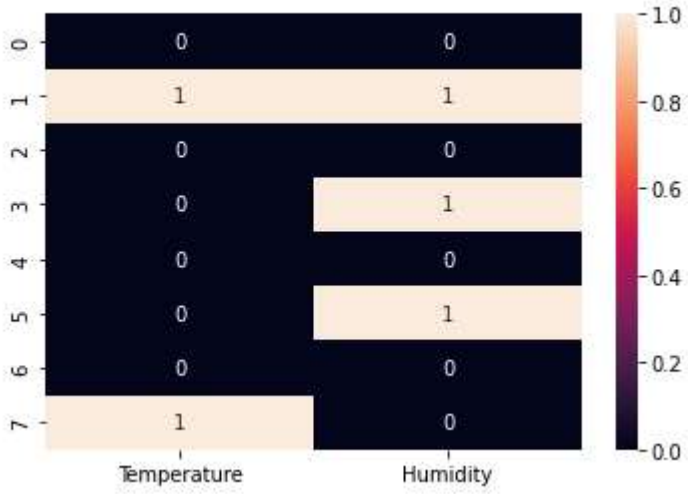
import seaborn as s
s.heatmap(df.isnull())

```
In [25]: s.heatmap(df.isnull())
Out[25]: <AxesSubplot:>
```



s.heatmap(df.isnull(),annot=True)

```
Out[27]: <AxesSubplot:>
```



df.dropna()

Out[28]:

	Temperature	Humidity
0	1.0	22.0
2	3.0	2.0
4	3.0	22.0
6	5.0	25.0

df.dropna(how='all')

Out[29]:

	Temperature	Humidity
0	1.0	22.0
2	3.0	2.0
3	2.0	NaN
4	3.0	22.0
5	4.0	NaN
6	5.0	25.0
7	NaN	27.0

In [30]:
```
df.fillna(0)
```

Out[30]:

	Temperature	Humidity
0	1.0	22.0
1	0.0	0.0
2	3.0	2.0
3	2.0	0.0
4	3.0	22.0
5	4.0	0.0
6	5.0	25.0
7	0.0	27.0

```
In [31]: df.fillna(method='ffill')
```

Out[31]:

	Temperature	Humidity
0	1.0	22.0
1	1.0	22.0
2	3.0	2.0
3	2.0	2.0
4	3.0	22.0
5	4.0	22.0
6	5.0	25.0
7	5.0	27.0

```
In [32]: df.fillna(method='bfill')
```

Out[32]:

	Temperature	Humidity
0	1.0	22.0
1	3.0	2.0
2	3.0	2.0
3	2.0	22.0
4	3.0	22.0
5	4.0	25.0
6	5.0	25.0
7	NaN	27.0

```
In [33]: df.interpolate(method='linear')
```

Out[33]:

	Temperature	Humidity
0	1.0	22.0
1	2.0	12.0
2	3.0	2.0
3	2.0	12.0
4	3.0	22.0
5	4.0	23.5
6	5.0	25.0
7	5.0	27.0

```
In [36]: df.fillna({'Temperature':888})
```

Out[36]:

	Temperature	Humidity
0	1.0	22.0
1	888.0	NaN
2	3.0	2.0
3	2.0	NaN
4	3.0	22.0
5	4.0	NaN
6	5.0	25.0
7	888.0	27.0

```
In [37]: df.fillna({'Humidity':999})
```

Out[37]:

	Temperature	Humidity
0	1.0	22.0
1	NaN	999.0
2	3.0	2.0
3	2.0	999.0
4	3.0	22.0
5	4.0	999.0
6	5.0	25.0
7	NaN	27.0

```
In [38]: df.Temperature.median()
```

Out[38]: 3.0

In [40]:
```
import math as m
```

In [42]:
```
median_temp=m.floor(df.Temperature.median())
print(median_temp)
df.fillna(median_temp)
```

```
3
```

Out[42]:

	Temperature	Humidity
0	1.0	22.0
1	3.0	3.0
2	3.0	2.0
3	2.0	3.0
4	3.0	22.0
5	4.0	3.0
6	5.0	25.0
7	3.0	27.0

6. Program to demonstrate linear regression (use1D array) for following data

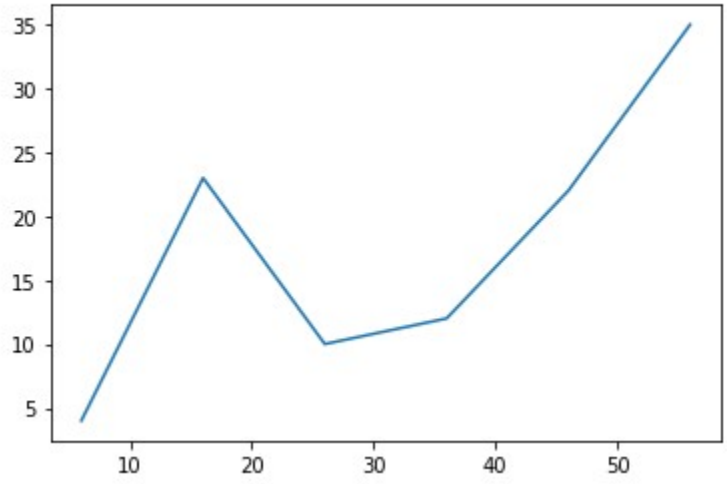
X	6	16	26	36	46	56
Y	4	23	10	12	22	35

```
import numpy as np
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot as plt
x=np.array([6,16,26,36,46,56]).reshape(-1,1)
x
```

```
Out[5]: array([[ 6],
               [16],
               [26],
               [36],
               [46],
               [56]])
```

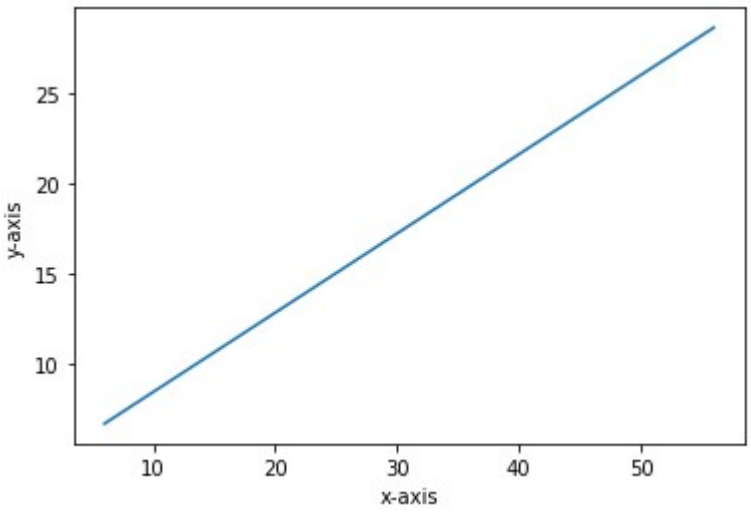
```
y=np.array([4,23,10,12,22,35])
plt.plot(x,y)
```

```
Out[7]: [<matplotlib.lines.Line2D at 0x1f760607d00>]
```



```
In [9]: m=LinearRegression().fit(x,y)

In [10]: print("coefficents,",m.score(x,y))
         coefficents, 0.5417910447761195

In [11]: print("intercept:",m.intercept_)
         intercept: 4.026666666666664

In [12]: print("slope:",m.coef_)
         slope: [0.44]

In [16]: y_predict=m.predict(x)
         print("result:",y_predict)

         result: [ 6.66666667 11.06666667 15.46666667 19.86666667 24.26666667 28.66666667]
```

```
import matplotlib.pyplot as plt
x=[6,16,26,36,46,56]
y=[6.66666667,11.06666667,15.46666667,19.86666667,24.26666667,28.66666667]
plt.plot(x,y)
plt.ylabel("y-axis")
plt.xlabel("x-axis")
plt.show()
```



7. Program to create a model for the following data. Use the created model to predict the price of an area given

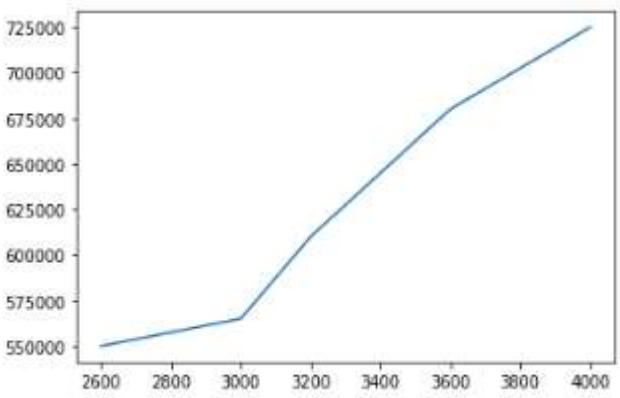
Area	price
2600	550000
3000	565000
3200	610000
3600	680000
4000	725000

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot as plt
#load data
df=pd.read_csv("a_price.csv")
df
```

```
In [12]: #load data
         df=pd.read_csv("a_price.csv")
         df

Out[12]:
```

	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	680000
4	4000	725000

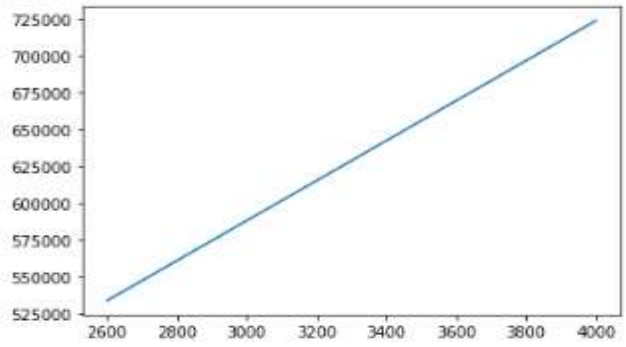
```
plt.plot(df.area,df.price)
plt.show()
```



#fit the data with model
reg=LinearRegression().fit(df[['area']],df.price)
data_predict=reg.predict(df[['area']])
print("result",data_predict)

```
In [36]: data_predict=reg.predict(df[['area']])
         print("result",data_predict)

         result [533664.38356164 587979.45205479 615136.98630137 669452.05479452
          723767.12328767]
```

```
In [37]: plt.plot(df['area'],data_predict)
         plt.show()
```




```
In [40]: area=int(input('enter area:'))
         data_predict=reg.predict([[area]])
         print('the price of given area is',data_predict)

         enter area:4500
         the price of given area is [791660.95890411]

         C:\Users\PG LAB\AppData\Roaming\Python\Python310\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature
         names, but LinearRegression was fitted with feature names
           warnings.warn(
```

8. Program to create a model for the following data(csv file). Apply data cleaning if neded. Use the model to predict the salary for the data (experience-12,test_Score-10,interview-10)

exp	test_score	interview_	salary
	8	9	50000
	8	6	45000
5	6	7	60000
2	10	10	65000
7	9	6	70000
3	7	10	62000
10		7	72000
11	7	8	80000

	A	B	C	D
1	exp	test_score	interview	salary
2		8	9	50000
3		8	6	45000
4	5	6	7	60000
5	2	10	10	65000
6	7	9	6	70000
7	3	7	10	62000
8	10		7	72000
9	11	7	8	80000
10				

In [41]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression
```

In [49]:
```
import numpy
df=pd.read_csv("interview.csv")
df
```

Out[49]:

	exp	test_score	interview_score	salary
0	NaN	8.0	9	50000
1	NaN	8.0	6	45000
2	5.0	6.0	7	60000
3	2.0	10.0	10	65000
4	7.0	9.0	6	70000
5	3.0	7.0	10	62000
6	10.0	NaN	7	72000
7	11.0	7.0	8	80000

In [50]:
```
import numpy as np
missing_values=[np.nan]
df=p.read_csv('interview.csv',na_values=missing_values)
```

In [51]:
```
df['exp']=df['exp'].fillna(0)
df
```

Out[51]:

	exp	test_score	interview_score	salary
0	0.0	8.0	9	50000
1	0.0	8.0	6	45000
2	5.0	6.0	7	60000
3	2.0	10.0	10	65000
4	7.0	9.0	6	70000
5	3.0	7.0	10	62000
6	10.0	NaN	7	72000
7	11.0	7.0	8	80000

In [52]:
```
test_median=df.test_score.median()
test_median
```

Out[52]: 8.0

In [55]:
```
df['test_score']=df['test_score'].fillna(test_median)
df
```

Out[55]:

	exp	test_score	interview_score	salary
0	0.0	8.0	9	50000
1	0.0	8.0	6	45000
2	5.0	6.0	7	60000
3	2.0	10.0	10	65000
4	7.0	9.0	6	70000
5	3.0	7.0	10	62000
6	10.0	8.0	7	72000
7	11.0	7.0	8	80000

```
In [59]: df
```

Out[59]:

	exp	test_score	interview_score	salary
0	0.0	8.0	9	50000
1	0.0	8.0	6	45000
2	5.0	6.0	7	60000
3	2.0	10.0	10	65000
4	7.0	9.0	6	70000
5	3.0	7.0	10	62000
6	10.0	8.0	7	72000
7	11.0	7.0	8	80000

```
In [57]: reg=LinearRegression().fit(df[['exp','test_score','interview_score']],df.salary)
```

```
In [62]: reg.predict([[12,6,8]])
```

```
C:\Users\PG LAB\AppData\Roaming\Python\Python310\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature
names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
Out[62]: array([80288.87918486])
```

```
In [58]: data_predict=reg.predict(df[['exp','test_score','interview_score']])
         print("result",data_predict)
```

```
result [52350.0727802  45734.35225619 58312.95487627 63872.63464338
 67270.74235808 61148.47161572 76069.1411936  79241.63027656]
```

```
In [65]: m=int(input('enter experience:'))
         n=int(input('enter test score:'))
         o=int(input('enter interview score:'))
         data_predict=reg.predict([[m,n,o]])
         print("Predicted Salary:",data_predict)
```

```
enter experience:12
enter test score:12
enter interview score:17
Predicted Salary: [111130.27656477]
```

9. Prediction using Logistic Regression on iris Dataset.

	A	B	C	D	E
1	sepal_len	sepal_wid	petal_len	petal_wid	species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa
15	4.3	3	1.1	0.1	setosa
16	5.8	4	1.2	0.2	setosa
17	5.7	4.4	1.5	0.4	setosa
18	5.4	3.9	1.3	0.4	setosa
19	5.1	3.5	1.4	0.3	setosa
20	5.7	3.8	1.7	0.3	setosa
21	5.1	3.8	1.5	0.3	setosa

10. Classification on iris dataset

```
In [8]: import pandas as pd
```

```
In [9]: from sklearn.model_selection import train_test_split
        from sklearn.neighbors import KNeighborsClassifier
```

```
In [19]: #Load data
         df=pd.read_csv('iris.csv')
         df
```

Out[19]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [20]:
```
#excluded class label

x=df.drop(columns=['Species'])
#class label column
y=df['Species']
x
```

Out[20]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

150 rows × 4 columns

In [21]:
```
#split the dataset in to training and testing
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30)
```

In [23]:
```
#fit the training data in to KNN model
kn=KNeighborsClassifier(n_neighbors=2)
kn.fit(x_train,y_train)
```

Out[23]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=2)
```

In [24]:
```
kn.predict([[2.1,3.2,4.1,3.0]])
```

```
C:\Users\PG LAB\AppData\Roaming\Python\Python310\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature
names, but KNeighborsClassifier was fitted with feature names
  warnings.warn(
```

Out[24]:
```
array(['Iris-versicolor'], dtype=object)
```

11. Clustering on Iris Dataset

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

```
#load data
df=pd.read_csv("iris.csv")
df
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
#fill NaN values with 0 for twp columns bcz clustering is doing based on these two columns
df['sepal_length']=df.sepal_length.fillna(0)
df['sepal_width']=df.sepal_width.fillna(0)
df
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
#two columns are assigned into x
x=df[['sepal_length','sepal_width']]
x.head()
x
```

	sepal_length	sepal_width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6
...	...	...
145	6.7	3.0
146	6.3	2.5
147	6.5	3.0
148	6.2	3.4
149	5.9	3.0

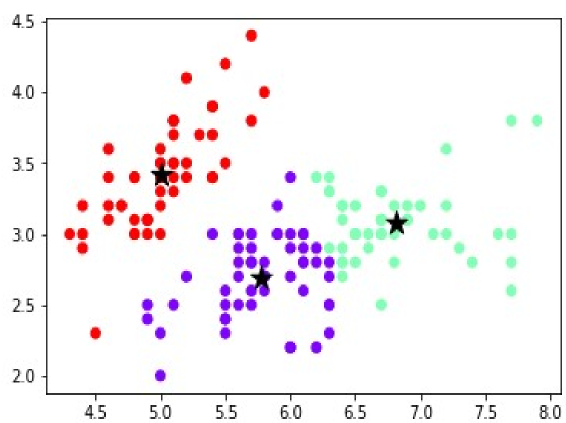
150 rows × 2 columns

```python
#now we are doing clustering using x
#model creation
m=KMeans(n_clusters=3,max_iter=50)

#fit data to model
m.fit(x)

#printing the cluster centers and labels
c=m.cluster_centers_
labels=m.labels_
print('cluster centers',c)
print('labels',labels)
c
```

```
cluster centers [[5.77358491 2.69245283]
 [6.81276596 3.07446809]
 [5.006      3.418     ]]
labels [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 1 1 1
 1 1 0 0 1 1 1 1 0 1 0 1 0 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1
 1 0]

array([[5.77358491, 2.69245283],
       [6.81276596, 3.07446809],
       [5.006     , 3.418     ]])
```

```python
#visualization

xaxis=df['sepal_length']
yaxis=df['sepal_width']
plt.scatter(xaxis,yaxis,c=labels,cmap='rainbow')

plt.scatter(c[0][0],c[0][1],marker='*',s=200,color='black')
plt.scatter(c[1][0],c[1][1],marker='*',s=200,color='black')
plt.scatter(c[2][0],c[2][1],marker='*',s=200,color='black')
#plt.scatter(c[3][0],c[3][1],marker='*',s=200,color='black')
```

```
<matplotlib.collections.PathCollection at 0x231dfa28b50>
```



DJANGO

Program: 1

Aim: Write a Django program to display 2 views

1. **Creating a Django project**

   C:\Python\Scripts>django-admin startproject SampleProject

2. **Open Sampleproject**

   C:\Python\Scripts>cd SampleProject

3. **Run Sampleproject**

   C:\Python\Scripts\SampleProject>python manage.py runserver

   Watching for file changes with StatReloader

   Performing system checks...

   System check identified no issues (0 silenced).

   You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

   Run 'python manage.py migrate' to apply them.

   March 16, 2023 - 10:48:34

   Django version 4.1, using settings 'SampleProject.settings'

   Starting development server at http://127.0.0.1:8000/

   Quit the server with CTRL-BREAK.

4. **Start app -home page**
   C:\Python\Scripts\SampleProject>python manage.py startapp homepage

5. **Homepage->view.py**
   from django.shortcuts import render
   from django.http import HttpResponse
   # Create your views here.
   def welcome(request):

```
    return HttpResponse("<h1> Welcome to My Project </h1>")
def contact(request):
    return HttpResponse("<h1> Contact details </h1>")
def hello(request):
    return HttpResponse("<h1> Hello</h1>")
```

6. **Homepage->create urls.py**

```
from django.urls import path
from . import views

urlpatterns = [
    path('home/', views.welcome),
    path('contact/', views.contact),
    path('hello/', views.hello),
]
```

7. **Connecting view and urls**

```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('homepage.urls')),
]
```

8. **Run the homepage**

```
C:\Python\Scripts\SampleProject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply
the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
March 16, 2023 - 11:19:04
Django version 4.1, using settings 'SampleProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

9. View of home page

10. View of contact page

Program: 2

Aim: Write a Django program to display static web page

1. **Creating a Django project**
   C:\Python\Scripts>django-admin startproject static_project1

2. **Open Project**
   C:\Python\Scripts>cd static_project1

3. **Create App**
   C:\Python\Scripts\static_project1>python manage.py startapp reg

4. **Reg(templates(index.html))**

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
button {
      background-color: #4CAF50;
            width: 100%;
          color: orange;
          padding: 15px;
          margin: 10px 0px;
          border: none;
          cursor: pointer;
          }
 form {
          border: 3px solid #f1f1f1;
          }
 input[type=text], input[type=password] {
          width: 100%;
          margin: 8px 0;
          padding: 12px 20px;
          display: inline-block;
```

```
        border: 2px solid green;
        box-sizing: border-box;
        }
 button:hover {
        opacity: 0.7;
        }
  .cancelbtn {
        width: auto;
        padding: 10px 18px;
        margin: 10px 5px;
        }


 .container {
        padding: 25px;
      background-color: rgb(193, 224, 235);
        }
</style>
</head>
<body>
        <center><h1> Student Login Form </h1></center>
        <form>
        <div class="container">
<label>Username : </label>
        <input type="text" placeholder="Enter Username" name="username" required>
<label>Password : </label>
        <input type="password" placeholder="Enter Password" name="password"
required>
        <button type="submit">Login</button>
        <input type="checkbox" checked="checked"> Remember me
        <button type="button" class="cancelbtn"> Cancel</button>
        Forgot <a href="#"> password? </a>
</div>
        </form>
</body>
</html>
```

5. **Static_project(settings.py)**

```
from pathlib import Path
```

```
# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-8huec$u8--^$adexfpt!xju723*jui#j-
5&*j44&3)!a%z0u9$'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'reg',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'static_project1.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'static_project1.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
```

```
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

6. **Reg_views.py**

```
from django.shortcuts import render
def login(request):
    return render(request, 'index.html')
```

7. **Reg_urls.py**

```
from django.urls import path
```

```
from . import views
urlpatterns = [
    path('login/', views.login),

]
```

8. Run Project

C:\Python\Scripts\static_project1>python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

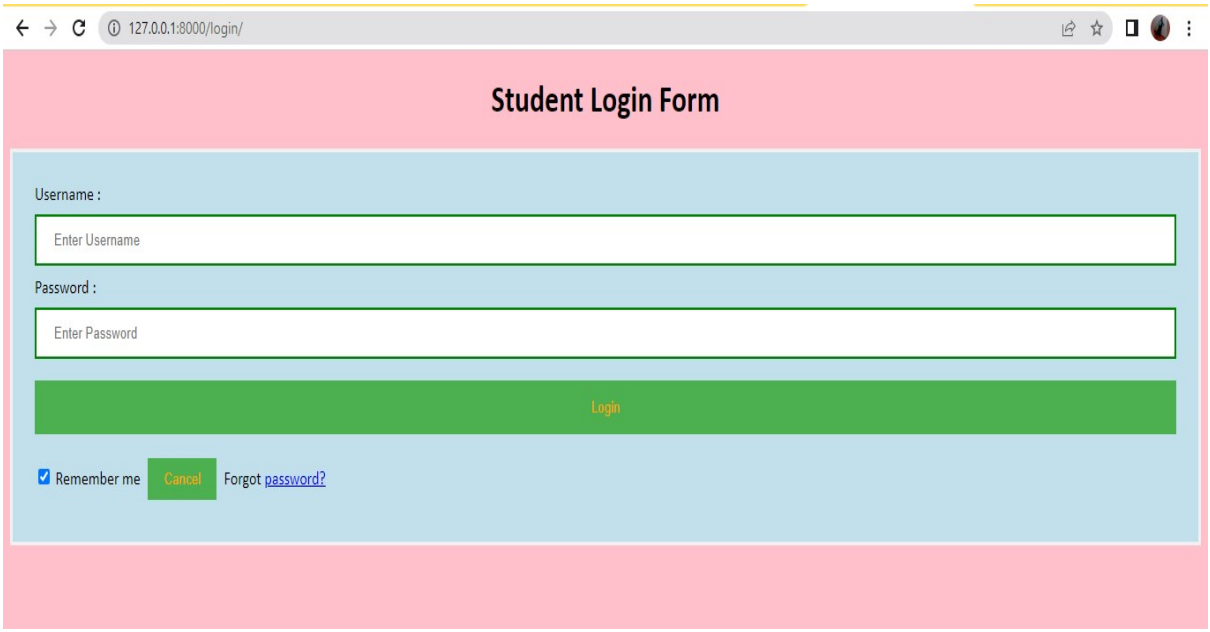
March 16, 2023 - 11:58:04

Django version 4.1, using settings 'static_project1.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

OUTPUT


127.0.0.1:8000/login/
Student Login Form
Username :
Enter Username
Password :
Enter Password
Login
Remember me
Cancel
Forgot password?