1.Write a python program to show use of multiple exception handling.

```
try:
    a=10
    b=0
    c="Jayashree"
    d=a+c
except TypeError:
    print("Type error occurs")
except ZeroDivisionError:
    print("Zero division error occurs")
except IndexError:
    print("Index error occurs")
```

o/p:
Type error occurs

2.Write a python function to check whether a number is perfect or not.

```
def Perfect_Number(Number):
    Sum = 0
    for i in range(1, Number):
        if(Number % i == 0):
            Sum = Sum + i
    return Sum
Number = int(input("Please Enter any number: "))
if (Number == Perfect_Number(Number)):
    print("\n %d is a Perfect Number" %Number)
else:
    print("\n %d is not a Perfect Number" %Number)
```

o/p:
Please Enter any number: 6

6 is a Perfect Number

Please Enter any number: 24

24 is not a Perfect Number

3.Write a python program to display only those words from the text file which contains three characters in it.

```
def read_file(filename):
    fp=open(filename,'r')
    data=fp.read()
    list1=data.split()
    for var in list1:
        if len(var)==3:
            print(var)
path=input("Enter the file name:")
read_file(path)
```

hello.txt content:
Ashoka Center for Business And Computer Studies
This is the python lab
First program

#steps for this file program:first go in your python folder->right click,select new and create text document(name as hello)->insert some characters which contain 3 characters->save the text document->come in IDLE python type the code and run
#after running input
Enter the file name:hello.txt
#output will be
#for
#And
#the
#lab

4. Write a python program to show how to use the else clause with try and except clauses.

```
def calculate_bmi(height,weight):
    return weight/height**2
def evaluate_bmi(bmi):
    if 18.5<=bmi<=24.9:
        return 'healthy'
    if bmi>=25:
        return 'overweight'
    return 'underweight'
def slip():
    try:
        height=float(input("Enter your height(meters):"))
        weight=float(input("Enter your weight(kilograms):"))
    except ValueError as error:
        print(error)
    else:
        bmi=round(calculate_bmi(height,weight),1)
        evaluation=evaluate_bmi(bmi)
        print(f"Your body mass index is {bmi}")
        print(f"This is considered {evaluation}!")
slip()
```

o/p:

```
Enter your height(meters):1.3
Enter your weight(kilograms):60
Your body mass index is 35.5
This is considered overweight!
```

5.Write a python program to count and display even and odd numbers of a List.

```
NumList = []
Even_count = 0
Odd_count = 0
Number = int(input("Please enter the Total Number of List Elements: "))
for i in range(1, Number + 1):
    value = int(input("Please enter the Value of %d Element : " %i))
    NumList.append(value)
for j in range(Number):
    if(NumList[j] % 2 == 0):
        Even_count = Even_count + 1
    else:
        Odd_count = Odd_count + 1
print("\nTotal Number of Even Numbers in this List =  ", Even_count)
print("Total Number of Odd Numbers in this List = ", Odd_count)
```

o/p:
Please enter the Total Number of List Elements: 6
Please enter the Value of 1 Element : 3
Please enter the Value of 2 Element : 4
Please enter the Value of 3 Element : 6
Please enter the Value of 4 Element : 7
Please enter the Value of 5 Element : 8
Please enter the Value of 6 Element : 9

Total Number of Even Numbers in this List = 3
Total Number of Odd Numbers in this List = 3

6.Write a python program to find the sum of items of a Dictionary.

```
def Sum(dic):
    sum=0
    for i in dic.values():
        sum=sum+i
    return sum
dic={'x':30,'y':40,'z':50}
print("Dictionary:",dic)
print("sum:",Sum(dic))
```

o/p:
Dictionary: {'x': 30, 'y': 40, 'z': 50}
sum: 120

7.Write a python program to show use of finally clause of exception handling.

```
try:
    numerator=10
    denominator=0
    result=numerator/denominator
    print(result)
except:
    print("Error:Denominator cannot be zero(0).")
finally:
    print("This is finally block.")
```

o/p:
Error:Denominator cannot be zero(0).
This is finally block.

8.Write a program that returns a list that contains only the elements that are common between the lists(without duplicates). Make sure your program works on two lists of different sizes.

```
one=[1,2,3,4,5,6,7,8,9,10,11,12,13]
two=[1,1,2,3,5,8,13,21,34,55,89]
both=[]
if len(one)<len(two):
    for i in one:
        if i in two and i not in both:
            both.append(i)
if len(one)>len(two):
    for i in two:
        if i in one and i not in both:
            both.append(i)
print(both)
```

o/p:
[1, 2, 3, 5, 8, 13]

9.Write a python program which accepts file name and word to be searched in file.Display line numbers which contain given words and total occurrences of it.

```
fname=input("Enter the file name:\t")
word=input("Enter the word to search:\t")
count=0
i=0
fp=open(fname,'r')
for line in fp:
    words=line.split()
    i=i+1
    for var in words:
        if(var==word):
            count=count+1
            print("Word\t",word,"is at line",i)
print("Number of occurrences of word are",count)
```

data.txt content:
Rahul
Jayashree
Diya
Jayashree
Jayesh
Ishan
Ved
Pune Ved
Mumbai Jayashree

#create text document in your folder where you are going to save your python program(name file as data and insert some values),write program in IDLE and run
#Input:
#Enter the file name: data.txt
#Enter the word to search: Jayashree
#output:
#Word Jayashree is at line 2
#Word Jayashree is at line 4
#Word Jayashree is at line 9
#Number of occurrences of word are 3

10. Write a python program to show how to raise an exception in python.

```
try:
   roll = int ( input ( "Please enter your roll number: " ) )
   if roll <= 0:
      raise ValueError ( "This is not a positive number!!" )
   print(num)
except ValueError as v:
   print("ValueError Exception thrown")
   print(v)
except:
   print("Roll number is valid")
print("\n\nOutside of try-except clauses.")
```

o/p:

- Please enter your roll number: 5
  Roll number is valid

  Outside of try-except clauses.

- Please enter your roll number: 0
  ValueError Exception thrown
  This is not a positive number!!

  Outside of try-except clauses.

11.Write a python program to display ‘n’ terms of Fibonacci series using recursion.

```
def recur_fibo(n):
    if n<=1:
        return n
    else:
        return(recur_fibo(n-1)+recur_fibo(n-2))
nterms=int(input("Please enter your number:"))
if nterms<=0:
    print("Please enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(recur_fibo(i))
```

o/p:

```
Please enter your number:5
Fibonacci sequence:
0
1
1
2
3
```

12.Write a python program to reverse each word of sentence of a file and also count total lines.

```
num_lines=0
with open("input.txt","r")as f:
    for line in f:
        num_lines+=1
with open("input.txt","r")as f:
    data=f.read()
    for line in f:
        num_lines+=1
new_data=""
for word in data.split():
    rev_word=word[::-1]
    new_data+=rev_word
    new_data+=""
print(new_data)
print("Number of lines:")
print(num_lines)

#create  text document as input.txt(insert values)
```

#write program
#run program
#output will be:
#eerhsayajluharayidhseyajnahsidevscbca
#Number of lines:
#7

13. Write a python program to show use of assert keyword.

```
def avg(marks):
    assert len(marks)!=0,"List is empty."
    return sum(marks)/len(marks)
mark2=[55,88,78,90,79]
print("Average of mark2:",avg(mark2))
mark1=[]
print("Average of mark1:",avg(mark1))
```

o/p:

```
Average of mark2: 78.0
Traceback (most recent call last):
  File "C:\Users\Lenovo\Desktop\python jayashree\exam.py", line 7, in <module>
    print("Average of mark1:",avg(mark1))
  File "C:\Users\Lenovo\Desktop\python jayashree\exam.py", line 2, in avg
    assert len(marks)!=0,"List is empty."
AssertionError: List is empty.
```

14.Write a python program to perform the following task.
    a. Calculate the factorial of a given number.
    b. Reverse the given number.

#factorial code:

```
num=int(input("Enter the number to calculate factorial:"))
factorial=1
if num<0:
    print("Sorry,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("Factorial of",num,"is",factorial)
```

o/p:

```
Enter the number to calculate factorial:5
Factorial of 5 is 120
```

Enter the number to calculate factorial:0
The factorial of 0 is 1
Enter the number to calculate factorial:-1
Sorry,factorial does not exist for negative numbers

#reverse number code:

```
num = int(input("Enter a number: "))
reversed_num=0
while num!=0:
        digit=num%10
        reversed_num=reversed_num*10+digit
        num//=10
print("Reversed number :"+str(reversed_num))
```

o/p:
Enter a number: 56
Reversed number :65

15.Write a python program which takes file name as input and print the lines after making only first character of each word in the sentence capitalized.

```
fo=open("hello1.txt","r")
totaldata=""
for data in fo:
    content=data.title()
    totaldata=totaldata+content
print(totaldata)
```

hello1.txt content:
i am jayashree kumawat
i am in msc
acbcs

#steps for this file program:first go in your python folder->right click,select new and create text document(name as hello1)->insert some characters which contain all small characters->save the text document->come in IDLE python type the code and run
#after running code
#output is:
#I Am Jayashree Kumawat
#I Am In Msc
#Acbcs

16.Write a python program to create a lambda function that adds 15 to a given number passed in as an argument.

```
result=lambda a:a+15
num=int(input("Enter the number:"))
print(result(num))
```

o/p:
Enter the number:20
35

17. Write a python program to print following pattern using loop.
A
B C
D E F
G H I J

```
rows=int(input("Enter the number of rows:"))
var=65
for i in range(rows):
    for j in range(i+1):
        print("\t",chr(var),end="")
        var += 1
    print()
```

o/p:
Enter the number of rows:4

```
        A
        B       C
        D       E       F
        G       H       I       J
```

18. Write a Python program to reverse the content of a file and store it in another file.

```
f1 = open("output1.txt", "w")
with open("file.txt", "r") as myfile:
    data = myfile.read()
data_1 = data[::-1]
f1.write(data_1)
f1.close()
```

#first create two text document in python folder where we will store the program,give input file name as file.txt and write some content and save it,now create another blank text document give name as output1.txt (dont write anything in output1.txt)

#write program ,save and run but ouput will not be on IDLE ,output will be reversed data in output1.txt(open this output file and see output)

file.txt content:
Jayashree kumawat
ashoka center for business and computer studies

output1.txt content:
seiduts retupmoc dna ssenisub rof retnec akohsa
tawamuk eerhsayaJ

19.Write a python program to show how to handle multiple exceptions.

```
try:
    a=10
    b=0
    c="abc"
    d=a/b
except ZeroDivisionError:
    print("Zero Division Error Occurs")
except IndexError:
    print("Index Error Occurs")
except TypeError:
    print("Type Error Occurs")
```

o/p:
Zero Division Error Occurs

20.Write a python program to display tables from m to n.
Example Input: m=3, n=7

Output: 3*1=3 4*1=4 ..... 7*1=7
3*2=6 4*2=8 ..... 7*2=14
.
.
.
3*10=30 4*10=40 ... 7*10=70

```
m=int(input("Enter the value of m:"))
n=int(input("Enter the value of n:"))
for i in range(1,11):
    for j in range(m,n+1):
        print(j,"*",i,"=",j*i,end="\t")
    print()
```

o/p:

```
Enter the value of m:3
Enter the value of n:7
3 * 1 = 3       4 * 1 = 4       5 * 1 = 5       6 * 1 = 6       7 * 1 = 7
3 * 2 = 6       4 * 2 = 8       5 * 2 = 10      6 * 2 = 12      7 * 2 = 14
3 * 3 = 9       4 * 3 = 12      5 * 3 = 15      6 * 3 = 18      7 * 3 = 21
3 * 4 = 12      4 * 4 = 16      5 * 4 = 20      6 * 4 = 24      7 * 4 = 28
3 * 5 = 15      4 * 5 = 20      5 * 5 = 25      6 * 5 = 30      7 * 5 = 35
3 * 6 = 18      4 * 6 = 24      5 * 6 = 30      6 * 6 = 36      7 * 6 = 42
3 * 7 = 21      4 * 7 = 28      5 * 7 = 35      6 * 7 = 42      7 * 7 = 49
3 * 8 = 24      4 * 8 = 32      5 * 8 = 40      6 * 8 = 48      7 * 8 = 56
3 * 9 = 27      4 * 9 = 36      5 * 9 = 45      6 * 9 = 54      7 * 9 = 63
3 * 10 = 30     4 * 10 = 40     5 * 10 = 50     6 * 10 = 60     7 * 10 = 70
```

21.Write a python program to accept directory name and print names of all files whose extension is '.txt' in the given directory.

```
import glob
print("Enter the extension(eg .txt,.html,.css etc):",end="")
e=input()
fileslist=[]
for file in glob.glob("*"+e):
    fileslist.append(file)
if len(fileslist)>0:
    print("\nList of all files with\""+e+"\"Extension:")
    for f in fileslist:
        print(f)
else:
    print("\n Not found with\""+e+"\"extension!")
```

o/p:

```
Enter the extension(eg .txt,.html,.css etc): .txt

List of all files with".txt"Extension:
data.txt
file.txt
foo.txt
hello.txt
hello1.txt
input.txt
output.txt
output1.txt
```

Steps for executing this program:
First create the folder ->name as exam(or as per your wish)->create some text document with .txt extension and write anything (or keep blank) ,save it in same folder only->write the python program and save in the exam folder where you had saved the text documents ->now run the program and you will be able to see your text documents name

22.Write a python program to calculate the cube of all numbers from 1 to n.

```
num=int(input("Enter the number : "))
for i in range(1,num+1):
    cb=i**3
    print("Cube of {0} is {1}".format(i,cb))
```

o/p:

```
Enter the number : 5
Cube of 1 is 1
Cube of 2 is 8
```

Cube of 3 is 27
Cube of 4 is 64
Cube of 5 is 125

23.Write a python program to display all prime numbers within given range.

```
start = int(input("Enter the lower bound: "))
stop = int(input("Enter the upper bound: "))
print("Prime numbers between", start, "and", stop, "are:")
for val in range(start, stop):
  if val>1:
    for i in range(2, val):
      if (val % i) == 0:
        break
    else:
      print(val, end=" ")
```

o/p:
Enter the lower bound: 1
Enter the upper bound: 50
Prime numbers between 1 and 50 are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

24.Write a python program to list only files from a directory and print files count.

```
import os
dir_path=r'C:\Users\Lenovo\Desktop\python jayashree'
print("Files and directories in'",dir_path,"':")
count=0
for path in os.listdir(dir_path):
    if os.path.isfile(os.path.join(dir_path,path)):
        print(path)
        count+=1
print("\n\nFile count:",count)
```

o/p:
Files and directories in' C:\Users\Lenovo\Desktop\python jayashree ':
1.py
13.py
2.py
3.py
4.py
5.py
6.py
7.py

class employee demo.py
data.txt
demo for class.py
exam.py
exception demo.py
file sample1.py
file.txt
foo.txt
frequency.py
function perfect number within range.py
hello.txt
hello1.txt
heterogram.py
hierarchical inheritance demo.py
hybrid inheritance demo.py
input.txt
insert list and pop.py
insert list.py
list comprehension.py
list to dictionary.py
multilevel inheritance demo.py
multiple inheritance demo.py
output.txt
output1.txt
QB 1.py
QB 2.py
QB 4.py
QB 5.py
QB 7.py
QB 8.py
QB 9.py
set.py
single inheritance demo.py
slip binary search.py
slip check key in dictionary.py
slip class rectangle.py
slip conversion of string.py
slip cube.py
slip even odd.py
slip factorial code.py
slip file capital first letter.py
slip file print 3 characters from text.py
slip file print occurences of word.py
slip file reverse and count.py
slip file reverse.py
slip finally clause.py
slip function perfect number.py

slip index odd values.py
slip less than 25.py
slip letters and digits count.py
slip multiple exception.py
slip pattern.py
slip prime number within range.py
slip raise exception.py
slip rectangle.py
slip repeated items in tuple.py
slip reverse number code.py
slip reverse string word by word using join and split.py
slip reversed string.py
slip set program to add elements.py
slip square using list comprension.py
slip swap strings.py
slip table.py
slip tuple check element.py
slip uppercase &lowercase.py
square list.py
string reverse.py
testfile
viva pattern prog.py
vowel count using set.py
walk demo.py
zip of greater string.py
zip.py

File count: 81

Steps for this program:
Go in any folder where lots of files are stored of any type
Basically start to select from desktop->This pc->select any folder ->In top , select and path will generate copy and paste that path in second line of code.

25.Write a python program to find square of given number using list comprehension.

```
list1=[]
list2=[]
element=int(input("Enter the number of elements in list:"))
for i in range(element):
    num=int(input("Enter the elements in list:"))
    list1.append(num)
list2=[i*i for i in list1]
print("Given list is:",list1)
print("Resultant list is:",list2)
```

o/p:
Enter the number of elements in list:5
Enter the elements in list:1
Enter the elements in list:2
Enter the elements in list:3
Enter the elements in list:4
Enter the elements in list:5
Given list is: [1, 2, 3, 4, 5]
Resultant list is: [1, 4, 9, 16, 25]

26.Write a python program which will find all such numbers which are divisible by 3 and not by 7 within given range m to n.

```
start_num=int(input("Enter the start number(m):"))
end_num=int(input("Enter the end number(n):"))
cnt=start_num
while cnt<=end_num:
    if cnt%3==0 and cnt % 7!=0:
        print(cnt,"is divisible by 3 and not by 7.")
    cnt+=1
```

o/p:
Enter the start number(m):1
Enter the end number(n):50
3 is divisible by 3 and not by 7.
6 is divisible by 3 and not by 7.
9 is divisible by 3 and not by 7.
12 is divisible by 3 and not by 7.
15 is divisible by 3 and not by 7.
18 is divisible by 3 and not by 7.
24 is divisible by 3 and not by 7.
27 is divisible by 3 and not by 7.
30 is divisible by 3 and not by 7.
33 is divisible by 3 and not by 7.
36 is divisible by 3 and not by 7.
39 is divisible by 3 and not by 7.
45 is divisible by 3 and not by 7.
48 is divisible by 3 and not by 7.

27.Write a python program to reverse each word of file and also count total lines.

```
num_lines=0
with open("input.txt","r")as f:
    for line in f:
        num_lines+=1
with open("input.txt","r")as f:
    data=f.read()
    for line in f:
        num_lines+=1
new_data=""
for word in data.split():
    rev_word=word[::-1]
    new_data+=rev_word
    new_data+=""
print(new_data)
print("Number of lines:")
print(num_lines)

#create  text document as input.txt(insert values)
#write program
#run program
#output will be:
#eerhsayajluharayidhseyajnahsidevscbca
#Number of lines:
#7
```

28.Write a python program which create a lambda function that multiplies argument x with argument y and print the result.

```
f=lambda a,b:a*b
x=int(input("Enter the value of x:"))
y=int(input("Enter the value of y:"))
print("Multiplication of",x,"and",y,"is:")
print(f(x,y))
```

o/p:

```
Enter the value of x:10
Enter the value of y:2
Multiplication of 10 and 2 is:
20
```

29.Write a python program to display all files in directory and subdirectories.

```
import os
path=r"C:\Users\Lenovo\Desktop\python jayashree"
filelist=[]
for root,dirs,files in os.walk(path):
    for file in files:
        filelist.append(os.path.join(root,file))
for name in filelist:
    print(name)


# on desktop create one folder(name as exam) insert some more folders and in sub-folders insert empty text documents
# Now insert some empty text documents in main folder and save your python program here in this main folder and in top path will be generated copy it and paste it on second line of code
# write program and execute
```

o/p:

```
C:\Users\Lenovo\Desktop\python jayashree\1.py
C:\Users\Lenovo\Desktop\python jayashree\13.py
C:\Users\Lenovo\Desktop\python jayashree\2.py
C:\Users\Lenovo\Desktop\python jayashree\3.py
C:\Users\Lenovo\Desktop\python jayashree\4.py
C:\Users\Lenovo\Desktop\python jayashree\5.py
C:\Users\Lenovo\Desktop\python jayashree\6.py
C:\Users\Lenovo\Desktop\python jayashree\7.py
C:\Users\Lenovo\Desktop\python jayashree\class employee demo.py
C:\Users\Lenovo\Desktop\python jayashree\data.txt
C:\Users\Lenovo\Desktop\python jayashree\demo for class.py
C:\Users\Lenovo\Desktop\python jayashree\exam.py
C:\Users\Lenovo\Desktop\python jayashree\exception demo.py
C:\Users\Lenovo\Desktop\python jayashree\file sample1.py
C:\Users\Lenovo\Desktop\python jayashree\file.txt
C:\Users\Lenovo\Desktop\python jayashree\foo.txt
C:\Users\Lenovo\Desktop\python jayashree\frequency.py
C:\Users\Lenovo\Desktop\python jayashree\function perfect number within range.py
C:\Users\Lenovo\Desktop\python jayashree\hello.txt
C:\Users\Lenovo\Desktop\python jayashree\hello1.txt
C:\Users\Lenovo\Desktop\python jayashree\heterogram.py
C:\Users\Lenovo\Desktop\python jayashree\hierarchical inheritance demo.py
C:\Users\Lenovo\Desktop\python jayashree\hybrid inheritance demo.py
C:\Users\Lenovo\Desktop\python jayashree\input.txt
C:\Users\Lenovo\Desktop\python jayashree\insert list and pop.py
C:\Users\Lenovo\Desktop\python jayashree\insert list.py
C:\Users\Lenovo\Desktop\python jayashree\list comprehension.py
C:\Users\Lenovo\Desktop\python jayashree\list to dictionary.py
```

C:\Users\Lenovo\Desktop\python jayashree\multilevel inheritance demo.py
C:\Users\Lenovo\Desktop\python jayashree\multiple inheritance demo.py
C:\Users\Lenovo\Desktop\python jayashree\output.txt
C:\Users\Lenovo\Desktop\python jayashree\output1.txt
C:\Users\Lenovo\Desktop\python jayashree\QB 1.py
C:\Users\Lenovo\Desktop\python jayashree\QB 2.py
C:\Users\Lenovo\Desktop\python jayashree\QB 4.py
C:\Users\Lenovo\Desktop\python jayashree\QB 5.py
C:\Users\Lenovo\Desktop\python jayashree\QB 7.py
C:\Users\Lenovo\Desktop\python jayashree\QB 8.py
C:\Users\Lenovo\Desktop\python jayashree\QB 9.py
C:\Users\Lenovo\Desktop\python jayashree\set.py
C:\Users\Lenovo\Desktop\python jayashree\single inheritance demo.py
C:\Users\Lenovo\Desktop\python jayashree\slip binary search.py
C:\Users\Lenovo\Desktop\python jayashree\slip check key in dictionary.py
C:\Users\Lenovo\Desktop\python jayashree\slip class rectangle.py
C:\Users\Lenovo\Desktop\python jayashree\slip conversion of string.py
C:\Users\Lenovo\Desktop\python jayashree\slip cube.py
C:\Users\Lenovo\Desktop\python jayashree\slip even odd.py
C:\Users\Lenovo\Desktop\python jayashree\slip factorial code.py
C:\Users\Lenovo\Desktop\python jayashree\slip file capital first letter.py
C:\Users\Lenovo\Desktop\python jayashree\slip file print 3 characters from text.py
C:\Users\Lenovo\Desktop\python jayashree\slip file print occurences of word.py
C:\Users\Lenovo\Desktop\python jayashree\slip file reverse and count.py
C:\Users\Lenovo\Desktop\python jayashree\slip file reverse.py
C:\Users\Lenovo\Desktop\python jayashree\slip finally clause.py
C:\Users\Lenovo\Desktop\python jayashree\slip function perfect number.py
C:\Users\Lenovo\Desktop\python jayashree\slip index odd values.py
C:\Users\Lenovo\Desktop\python jayashree\slip less than 25.py
C:\Users\Lenovo\Desktop\python jayashree\slip letters and digits count.py
C:\Users\Lenovo\Desktop\python jayashree\slip multiple exception.py
C:\Users\Lenovo\Desktop\python jayashree\slip pattern.py
C:\Users\Lenovo\Desktop\python jayashree\slip prime number within range.py
C:\Users\Lenovo\Desktop\python jayashree\slip raise exception.py
C:\Users\Lenovo\Desktop\python jayashree\slip rectangle.py
C:\Users\Lenovo\Desktop\python jayashree\slip repeated items in tuple.py
C:\Users\Lenovo\Desktop\python jayashree\slip reverse number code.py
C:\Users\Lenovo\Desktop\python jayashree\slip reverse string word by word using join and split.py
C:\Users\Lenovo\Desktop\python jayashree\slip reversed string.py
C:\Users\Lenovo\Desktop\python jayashree\slip set program to add elements.py
C:\Users\Lenovo\Desktop\python jayashree\slip square using list comprension.py
C:\Users\Lenovo\Desktop\python jayashree\slip swap strings.py
C:\Users\Lenovo\Desktop\python jayashree\slip table.py
C:\Users\Lenovo\Desktop\python jayashree\slip tuple check element.py
C:\Users\Lenovo\Desktop\python jayashree\slip uppercase &lowercase.py
C:\Users\Lenovo\Desktop\python jayashree\square list.py
C:\Users\Lenovo\Desktop\python jayashree\string reverse.py

C:\Users\Lenovo\Desktop\python jayashree\testfile
C:\Users\Lenovo\Desktop\python jayashree\viva pattern prog.py
C:\Users\Lenovo\Desktop\python jayashree\vowel count using set.py
C:\Users\Lenovo\Desktop\python jayashree\walk demo.py
C:\Users\Lenovo\Desktop\python jayashree\zip of greater string.py
C:\Users\Lenovo\Desktop\python jayashree\zip.py
C:\Users\Lenovo\Desktop\python jayashree\New folder\New Text Document.txt

30.Write a python program to delete repeated lines from a file.

```
outputFile=open("result.txt","w")
inputFile=open("inputs.txt","r")
lines=set()
print("Given Input file content is:")
for line in inputFile:
    print(line)
    if line not in lines:
        outputFile.write(line)
        lines.add(line)
inputFile.close()
outputFile.close()
outputFile=open("result.txt","r")
print("\n\nGiven output file content is:")
for line in outputFile:
    print(line)
```

#create one new folder in which you will save your python program
#create 2 text documents in same folder named as inputs.txt(insert records with repeated lines) and result.txt(keep it empty)

o/p:
Given Input file content is:
Hello

World of Python

Hello

How are you

Hello

World

Given output file content is:

Hello

World of Python

How are you

World

31.Write a python program to find the repeated items of a tuple.

```
list1=[int(x) for x in input("Enter the value:").split()]
print("Given list is:\t",list1)
tup=tuple(list1)
print("Given tuple is:\t",tup)
print("Following are repeated elements in given tuple:")
for i in range(len(tup)):
    for j in range(i+1,len(tup)):
        if tup[i]==tup[j]:
            print(tup[i])
```

o/p:
Enter the value:3 4 3 5 4 6 5
Given list is: [3, 4, 3, 5, 4, 6, 5]
Given tuple is: (3, 4, 3, 5, 4, 6, 5)
Following are repeated elements in given tuple:
3
4
5

32.Write a python program with user defined function which accept long string containing multiple words and it return same string with the words in backwards order.
Example: Input= “I am Msc student” then output = ”student Msc am I”

```
def reverse_string_words(text):
    for line in text.split('\n'):
        return(' '.join(line.split()[::-1]))
print(reverse_string_words("I am Msc student"))
print(reverse_string_words("Studying in ACBCS"))
```

o/p:
student Msc am I
ACBCS in Studying

33.Define a class Employee having members – id, name, department, salary. Create a subclass called —Manager with member bonus. Define methods accept and display in both the classes. Create n objects of the Manager class and display the details of the manager having the maximum total salary (salary + bonus).

```
class Employee:
    def __init__(self, emp_id, name, department, salary):
        self.id = emp_id
        self.name = name
        self.department = department
        self.salary = salary
    def accept(self):
        self.id = input("Enter ID: ")
        self.name = input("Enter Name: ")
        self.department = input("Enter Department: ")
        self.salary = float(input("Enter Salary: "))
    def display(self):
        print(f"ID: {self.id}")
        print(f"Name: {self.name}")
        print(f"Department: {self.department}")
        print(f"Salary: {self.salary}")
class Manager(Employee):
    def __init__(self, emp_id, name, department, salary, bonus):
        super().__init__(emp_id, name, department, salary)
        self.bonus = bonus
    def accept(self):
       super().accept()
       self.bonus = float(input("Enter Bonus: "))
    def display(self):
        super().display()
        print(f"Bonus: {self.bonus}")
n = int(input("Enter the number of managers: "))
managers = []
for i in range(n):
    print(f"Enter details for Manager {i + 1}:")
    manager = Manager(0, '', '', 0, 0)  # Initializing with default values
    manager.accept()
    managers.append(manager)
max_salary_manager = max(managers, key=lambda x: x.salary + x.bonus)
print("\nManager with the maximum total salary:")
max_salary_manager.display()
```

o/p:

```
Enter the number of managers: 2
Enter details for Manager 1:
Enter ID: 1
```

```
Enter Name: abc
Enter Department: msc
Enter Salary: 30000
Enter Bonus: 4000
Enter details for Manager 2:
Enter ID: 2
Enter Name: xyz
Enter Department: bsc
Enter Salary: 45000
Enter Bonus: 5000

Manager with the maximum total salary:
ID: 2
Name: xyz
Department: bsc
Salary: 45000.0
Bonus: 5000.0
```

34.Write a python program to find the length of a set. (Don't use built in function len)

```
intSet=set()
for i in range(1,8):
    value=int(input("Enter the %d Set Value="%i))
    intSet.add(value)
print("Set Items=",intSet)
length=0
for i in intSet:
    length+=1
print("Set Length=",length)
```

o/p:

```
Enter the 1 Set Value=1
Enter the 2 Set Value=2
Enter the 3 Set Value=3
Enter the 4 Set Value=4
Enter the 5 Set Value=5
Enter the 6 Set Value=6
Enter the 7 Set Value=7
Set Items= {1, 2, 3, 4, 5, 6, 7}
Set Length= 7
```

35.Write a python program that accepts a sentence and calculate the number of uppercase letters and lowercase letters

```
sentence=input("Enter the sentence:")
lower=0
upper=0
for c in sentence:
    if c.isupper():
        upper+=1
    elif c.islower():
        lower+=1
    else:
        pass
print("Given sentence is:",sentence)
print("Number of uppercase letters is:",upper)
print("Number of lowercase letters is:",lower)
```

o/p:

```
Enter the sentence:Ashoka Center For Business And Computer Studies
Given sentence is: Ashoka Center For Business And Computer Studies
Number of uppercase letters is: 7
Number of lowercase letters is: 34
```

36.Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area and perimeter. Display area and perimeter. Also delete the object.

```
class Rectangle():
    def __init__(self,l,w):
        self.length=l
        self.width=w
    def r_area(self):
        return self.length*self.width
    def r_peri(self):
        return 2*(self.length+self.width)
l=int(input("Enter the length:"))
w=int(input("Enter the width:"))
r_obj=Rectangle(l,w)
print("Area of rectangle is:",r_obj.r_area())
print("Perimeter of rectangle is:",r_obj.r_peri())
del r_obj
print(r_obj)
```

o/p:

```
Enter the length:10
Enter the width:15
Area of rectangle is: 150
Perimeter of rectangle is: 50
Traceback (most recent call last):
  File "C:\Users\Lenovo\Desktop\python jayashree\slip rectangle.py", line 15, in <module>
    print(r_obj)
NameError: name 'r_obj' is not defined
```

36.Write a python program to accept n elements in a set and find the length of a set, maximum, minimum value and the sum of values in a set.

```
import math
intSet=set()
n=int(input("Enter the number of elements in the set:"))
for i in range(1,n+1):
    value=int(input("Enter the %d Set Value="%i))
    intSet.add(value)
print("Set Items=",intSet)
length=0
maximum=0
minimum=math.inf
sumSet=0
for i in intSet:
    length+=1
    sumSet+=i
    if maximum<i:
        maximum=i
    if minimum>i:
        minimum=i
print("Set length=",length)
print("Maximum in set=",maximum)
print("Minimum in set=",minimum)
print("Sum of Set Elements=",sumSet)
```

```
o/p:
Enter the number of elements in the set:6
Enter the 1 Set Value=1
Enter the 2 Set Value=2
Enter the 3 Set Value=3
Enter the 4 Set Value=4
Enter the 5 Set Value=5
Enter the 6 Set Value=6
Set Items= {1, 2, 3, 4, 5, 6}
Set length= 6
```

Maximum in set= 6
Minimum in set= 1
Sum of Set Elements= 21

37.Write a python program that accepts a sentence and calculate the number of letters and digits in it.

```
sentence=input("Enter the sentence:")
d=l=0
for c in sentence:
    if c.isdigit():
        d=d+1
    elif c.isalpha():
        l=l+1
    else:
        pass
print("Letters are:", l)
print("Digits are:", d)
```

o/p:
Enter the sentence:Ashoka Center For Business And Computer Studies 422013
Letters are: 41
Digits are: 6

38.Write a python program to create a class Circle and compute the area and the circumference of the Circle. (Use parameterized constructor).

```
class Circle:
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return 3.14159 * self.radius ** 2
    def circumference(self):
        return 2 * 3.14159 * self.radius          # Using pi as 3.14159 for calculation
radius = float(input("Enter the radius of the circle: "))
new_circle = Circle(radius)
print("Area of the circle:", new_circle.area())
print("Circumference of the circle:", new_circle.circumference())
```

Enter the radius of the circle: 5
Area of the circle: 78.53975
Circumference of the circle: 31.4159

39.Write a program which checks whether given element exists within a tuple.

```
n=int(input("Enter length of tuple:"))
numbers=[]
for _ in range(n):
    temp=int(input("Enter List Item(Integer):"))
    numbers.append(temp)
t=tuple(numbers)
print(t)
item=int(input("Enter tuple item to search:"))
if item in t:
    print("Item is present in tuple")
else:
    print("Item is not present in tuple")
```

o/p:

```
Enter length of tuple:5
Enter List Item(Integer):2
Enter List Item(Integer):3
Enter List Item(Integer):4
Enter List Item(Integer):5
Enter List Item(Integer):6
(2, 3, 4, 5, 6)
Enter tuple item to search:4
Item is present in tuple

Enter length of tuple:3
Enter List Item(Integer):4
Enter List Item(Integer):5
Enter List Item(Integer):6
(4, 5, 6)
Enter tuple item to search:2
Item is not present in tuple
```

40.Write a Python program to find the greatest common divisor (gcd) of two integers.

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=int(input("Enter the value:"))
b=int(input("Enter the value:"))
print("The gcd of",a,"and",b,"is:",end=" ")
print(gcd(a,b))
```

o/p:
Enter the value:2
Enter the value:4
The gcd of 2 and 4 is: 2

41.Define a class Student having members – rollno, name, age, gender. Create a subclass called —Test with member marks of 3 subjects. Create three objects of the Test class and display all the details of the student with percentage.

```
class student:
    marks=[]
    def getData(self,name,rn,age,gender,m1,m2,m3):
        student.name=name
        student.rn=rn
        student.age=age
        student.gender=gender
        student.marks.append(m1)
        student.marks.append(m2)
        student.marks.append(m3)
    def displayData(self):
        print("Name is : ",student.name)
        print("Roll No. is :  ",student.rn)
        print("Age is  :",student.age)
        print("Gender is  :   ",student.gender)
        print("Marks are : ",student.marks)
        print("Total marks are : ",self.total())
        print("Percentage is : ",round(self.per(),2))
    def test(self):
        print("Marks in 1st subject :",student.marks[0])
        print("Marks in 2nd subject :",student.marks[1])
        print("Marks in 3rd subject :",student.marks[2])
    def total(self):
        return (m1+m2+m3)
    def per(self):
        return ((m1+m2+m3)/3)
name=input("Enter the name of student:")
rn=input("Enter the roll number:")
age=input("Enter the age:")
gender=input("Enter the gender:")
m1=int(input("Enter the marks in first subject:"))
m2=int(input("Enter the marks in second subject:"))
m3=int(input("Enter the marks in third subject:"))
s1=student()
s1.getData(name,rn,age,gender,m1,m2,m3)
s1.test()
s1.displayData()
```

o/p:

```
Enter the name of student:jayashree
Enter the roll number:7
Enter the age:22
Enter the gender:female
Enter the marks in first subject:30
Enter the marks in second subject:40
Enter the marks in third subject:50
Marks in 1st subject : 30
Marks in 2nd subject : 40
Marks in 3rd subject : 50
Name is :  jayashree
Roll No. is :   7
Age is  : 22
Gender is  :    female
Marks are :  [30, 40, 50]
Total marks are :  120
Percentage is :  40.0
```

42.Write a python program to find the repeated items of a tuple.

```
t=[int(x) for x in input("Enter the value:").split()]
t=tuple(t)
print("Repeated values:")
for i in range(0,len(t)):
    for j in range(i+1,len(t)):
        if t[i]==t[j]:
            print(t[i],end=" ")
```

o/p:

```
Enter the value:4 5 4 6 5 7 8
Repeated values:
4 5
```

43.Write a python program to accept string and remove the characters which have odd index values of a given string using user defined function.

```
def removeOddString(strng):
    tempstng=" "
    for charindex in range(len(strng)):
        if charindex%2==0:
            tempstng=tempstng+strng[charindex]
    return tempstng
givenstring=input("Enter any random string:")
print("The given string before modification is:",givenstring)
resstring=removeOddString(givenstring)
print("The given string after modification is:",resstring)
```

o/p:
Enter any random string:Jayashree
The given string before modification is: Jayashree
The given string after modification is: Jysre

44.Define a class Person having members – name, address. Create a subclass called —Employee with member staff id, salary. Create ‘n’ objects of the Employee class and display all the details of the Employee.

```
class person:
    def __init__(self,name,address):
        self.empname=name
        self.address=address
class employee(person):
    def __init__(self,name,address,salary):
        super(). __init__(name,address)
        self.salary=salary
    def display(self):
        print('name : {}\taddress : {}\tsalary : {}'.format(self.empname,self.address,self.salary))
n=int(input("Enter how many employees:"))
s=[]
for i in range(0,n):
    x=input("Enter object name:")
    s.append(x)
    print(s)
for j in range(0,n):
    print("\nEnter details of employee:", j+1)
    name1=input("Enter name:")
    address=input("Enter address:")
    salary=int(input("Enter salary:"))
    s[j]=employee(name1,address,salary)
print("\nDetails of Employee are:")
for j in range(0,n):
```

```
    s[j].display()
```

o/p:

```
Enter how many employees:2
Enter object name:e1
['e1']
Enter object name:e2
['e1', 'e2']

Enter details of employee: 1
Enter name:jayashree
Enter address:gangapur road
Enter salary:25000

Enter details of employee: 2
Enter name:diya
Enter address:panchavati
Enter salary:24000

Details of Employee are:
name : jayashree        address : gangapur road         salary : 25000
name : diya     address : panchavati  salary : 24000
```

45.Write a python program that prints out all the elements of the list that are less than 25.

```
list1=[10,50,60,80,20,15]
print("Elements in this list are :")
print(list1)
print("Elements in this list less than 25 are:")
for i in range(len(list1)):
    if list1[i]<25:
        print(list1[i])
```

o/p:

```
Elements in this list are :
[10, 50, 60, 80, 20, 15]
Elements in this list less than 25 are:
10
20
15
```

46.Create a class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

```
class Circle():
    def __init__(self,r):
        self.radius=r
    def area(self):
        return self.radius**2*3.14
    def perimeter(self):
        return 2*self.radius*3.14
NewCircle=Circle(8)
print(NewCircle.area())
print(NewCircle.perimeter())
```

o/p:
200.96
50.24

47. For given a .txt file that has a list of a bunch of names, count how many of each name there are in the file and print count.

```
text=open("test.txt","r")
d=dict()
for line in text:
    line=line.strip()
    line=line.lower()
    words=line.split(" ")
    for word in words:
        if word in d:
            d[word]=d[word]+1
        else:
            d[word]=1
for key in list(d.keys()):
    print(key,":",d[key])

# in your program folder where you will save the program,create one
# text document named as "test" and insert some values and save
```

o/p:
jayashree : 2
diya : 2
rahul : 2
jayesh : 2

test.txt content:
jayashree
diya
jayashree
diya
rahul
jayesh
rahul
jayesh

48.Write a python program which reverse given string and displays both original and reversed string. (Don't use built-in function)

```
my_str=input("Please enter your own string:")
str=' '
for i in my_str:
    str=i+str
print("\nThe original string is:",my_str)
print("The reversed string is:",str)
```

o/p:
Please enter your own string: jayesh

The original string is: jayesh
The reversed string is: hseyaj

49.Write a python program to implement binary search to search the given element using function.

```
def binary_search(arr,a,low,high):
    while low<=high:
        mid=low+(high-low)//2
        if arr[mid]==a:
            return mid
        elif array[mid]<a:
            low=mid+1
        else:
            high=mid-1
    return -1
arr=[1,2,3,4,5,6,7]
a=4
print("The given array is:",arr)
print("Element to be found is:",a)
index=binary_search(arr,a,0,len(arr)-1)
if index !=-1:
    print("The index of the element is  "+str(index))
else:
    print("Element not found")
```

o/p:
The given array is: [1, 2, 3, 4, 5, 6, 7]
Element to be found is: 4
The index of the element is 3

50.Write a python program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

```
one=[1,2,3,4,5,6,7,8,9,10,11,12,13]
two=[1,1,2,3,5,8,13,21,34,55,89]
both=[]
if len(one)<len(two):
    for i in one:
        if i in two and i not in both:
            both.append(i)
if len(one)>len(two):
    for i in two:
        if i in one and i not in both:
            both.append(i)
print(both)
```

o/p:
[1, 2, 3, 5, 8, 13]

51.Write a python program to count the number of characters in a string without using any built-in function.

```
def findLength(string):
    count=0
    for i in string:
        count+=1
    return count
string=input("Enter the string:")
print("Number of characters in the string are\t",findLength(string))
```

o/p:
Enter the string:Jayashree Kumawat
Number of characters in the string are 17

Space is also considered in count

52.Define a class Person having members – name, address. Create a subclass called —Employee with member staff id, salary. Create ‘n’ objects of the Employee class and display all the details of highest salaried employee.

```
class Person:
    def __init__(self, name, address):
        self.name = name
        self.address = address
class Employee(Person):
    def __init__(self, name, address, staff_id, salary):
        super().__init__(name, address)
        self.staff_id = staff_id
        self.salary = salary
def find_highest_salaried_employee(employees):
    if not employees:
        return None
    highest_salary_employee = max(employees, key=lambda emp: emp.salary)
    return highest_salary_employee
# Creating 'n' Employee objects
n = 2  # Number of employees
employees_list = []
for i in range(n):
    name = input(f"Enter name of employee {i+1}: ")
    address = input(f"Enter address of employee {i+1}: ")
    staff_id = input(f"Enter staff ID of employee {i+1}: ")
    salary = float(input(f"Enter salary of employee {i+1}: "))
    employee = Employee(name, address, staff_id, salary)
    employees_list.append(employee)
# Find the details of the highest salaried employee
highest_salaried_employee = find_highest_salaried_employee(employees_list)
if highest_salaried_employee:
    print("\nDetails of the highest salaried employee:")
    print(f"Name: {highest_salaried_employee.name}")
    print(f"Address: {highest_salaried_employee.address}")
    print(f"Staff ID: {highest_salaried_employee.staff_id}")
    print(f"Salary: {highest_salaried_employee.salary}")
else:
    print("No employees found.")
```

o/p:

```
Enter name of employee 1: jayashree
Enter address of employee 1: nashik
Enter staff ID of employee 1: 101
Enter salary of employee 1: 25000
```

```
Enter name of employee 2: diya
Enter address of employee 2: pune
Enter staff ID of employee 2: 102
Enter salary of employee 2: 30000

Details of the highest salaried employee:
Name: diya
Address: pune
Staff ID: 102
Salary: 30000.0
```

53.Write a python program to check if a given key already exists in a dictionary. If key exists replace with another key/value pair.

```
def checkKey(dic,key):
    print("Given Dictionary is:",dic)
    if key in dic.keys():
        print("Key Present,",end=" ")
        print("Value=",dic[key])
        del(dic[key])
        k=(input("Enter new key:"))
        v=int(input("Enter new value:"))
        dic[k]=v
        print("New dictionary is :",dic)
    else:
        print("Key is not present")
dic={'a':100,'b':200,'c':300}
key='b'
checkKey(dic,key)
```

```
o/p:
Given Dictionary is: {'a': 100, 'b': 200, 'c': 300}
Key Present, Value= 200
Enter new key:d
Enter new value:400
New dictionary is : {'a': 100, 'c': 300, 'd': 400}
```

54.Write a python program to get a single string from two given strings and swap the first two characters of each string.
Sample String: 'abc', 'xyz'
Expected Output: xycabz

```
a=input("Enter first string:")
b=input("Enter second string:")
x=a[0:2]
a=a.replace(a[0:2],b[0:2])
```

```
b=b.replace(b[0:2],x)
print(a,b)
```

o/p:

```
Enter first string:jayashree
Enter second string:kumawat
kuyashree jamawat
```

55.Define a class Person having members – name, address. Create a subclass called —Employee with members staff id, salary. Create ‘n’ objects of the Employee class and display all the details of the Employee.

```
class person:
    def __init__(self,name,address):
        self.empname=name
        self.address=address
class employee(person):
    def __init__(self,name,address,salary):
        super(). __init__(name,address)
        self.salary=salary
    def display(self):
        print('name : {}\taddress : {}\tsalary : {}'.format(self.empname,self.address,self.salary))
n=int(input("Enter how many employees:"))
s=[]
for i in range(0,n):
    x=input("Enter object name:")
    s.append(x)
    print(s)
for j in range(0,n):
    print("\nEnter details of employee:", j+1)
    name1=input("Enter name:")
    address=input("Enter address:")
    salary=int(input("Enter salary:"))
    s[j]=employee(name1,address,salary)
print("\nDetails of Employee are:")
for j in range(0,n):
    s[j].display()
```

o/p:

```
Enter how many employees:2
Enter object name:e1
['e1']
Enter object name:e2
['e1', 'e2']

Enter details of employee: 1
Enter name:abc
```

```
Enter address:nashik
Enter salary:2000

Enter details of employee: 2
Enter name:xyz
Enter address:pune
Enter salary:3000

Details of Employee are:
name : abc      address : nashik        salary : 2000
name : xyz      address : pune          salary : 3000
```

56.Write a python program to create a tuple of n numbers and print maximum, minimum,and sum of elements in a tuple. (Don't use built-in functions)

```
import math
tup=tuple()
n=int(input("Enter the number of elements in the tuple:"))
for i in range(1,n+1):
    value=int(input("Enter the %d set value="%i))
    tup=tup+(value,)
print("Tuple Items=",tup)
length=0
maximum=0
minimum=math.inf
sumtup=0
for i in tup:
    length+=1
    sumtup+=i
    if maximum<i:
        maximum=i
    if minimum>i:
        minimum=i
print("Tuple Length=",length)
print("Maximum in Tuple=",maximum)
print("Minimum in Tuple=",minimum)
print("Sum of tuple elements=",sumtup)
```

```
o/p:
Enter the number of elements in the tuple:5
Enter the 1 set value=3
Enter the 2 set value=5
Enter the 3 set value=6
Enter the 4 set value=1
Enter the 5 set value=2
Tuple Items= (3, 5, 6, 1, 2)
```

Tuple Length= 5
Maximum in Tuple= 6
Minimum in Tuple= 1
Sum of tuple elements= 17

57.Write a python program to accept and convert string in uppercase or vice versa.

```
str1=input("Enter string:")
newStr=" "
for i in range(0,len(str1)):
    if str1[i].islower():
        newStr+=str1[i].upper()
    elif str1[i].isupper():
        newStr+=str1[i].lower()
    else:
        newStr+=str1[i]
print("String after case conversion:"+newStr)
```

o/p:
Enter string:Jayashree
String after case conversion: jAYASHREE

58.Write a python program to create a class Calculator with basic calculator operations (addition,subtraction,division,multiplication,remainder).

```
class cal():
    def __init__(self,a,b):
        self.a=a
        self.b=b
    def add(self):
        return self.a+self.b
    def mul(self):
        return self.a*self.b
    def div(self):
        return  self.a/self.b
    def sub(self):
        return self.a-self.b
    def rem(self):
        return self.a%self.b
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
obj=cal(a,b)
choice=1
while choice!=0:
    print("0.Exit")
    print("1.Add")
    print("2.Subtraction")
```

```
    print("3.Multiplication")
    print("4.Division")
    print("5.Remainder")
    choice=int(input("Enter choice:"))
    if choice==1:
        print("Result:",obj.add())
    elif choice==2:
        print("Result:",obj.sub())
    elif choice==3:
        print("Result:",obj.mul())
    elif choice==4:
        print("Result:",round(obj.div(),2))
    elif choice==5:
        print("Result:",obj.rem())
    elif choice==0:
        print("Exiting")
    else:
        print("Invalid choice")
print()
```

o/p:

```
Enter first number:2
Enter second number:3
0.Exit
1.Add
2.Subtraction
3.Multiplication
4.Division
5.Remainder
Enter choice:1
Result: 5
0.Exit
1.Add
2.Subtraction
3.Multiplication
4.Division
5.Remainder
Enter choice:2
Result: -1
0.Exit
1.Add
2.Subtraction
3.Multiplication
4.Division
5.Remainder
Enter choice:3
Result: 6
```

```
0.Exit
1.Add
2.Subtraction
3.Multiplication
4.Division
5.Remainder
Enter choice:4
Result: 0.67
0.Exit
1.Add
2.Subtraction
3.Multiplication
4.Division
5.Remainder
Enter choice:5
Result: 2
0.Exit
1.Add
2.Subtraction
3.Multiplication
4.Division
5.Remainder
Enter choice:0
Exiting
```

59.Write a python program to perform operations on sets which includes union of two sets,an intersection of sets, set difference and a symmetric difference.

```
E={0,2,4,6,8}
N={1,2,3,4,5}
```

```
print("Union of E and N is:",E|N)
print("Intersection of E and N is:",E&N)
print("Difference of E and N is:",E-N)
print("Symmetric difference of E and N is:",E^N)
```

o/p:

```
Union of E and N is: {0, 1, 2, 3, 4, 5, 6, 8}
Intersection of E and N is: {2, 4}
Difference of E and N is: {0, 8, 6}
Symmetric difference of E and N is: {0, 1, 3, 5, 6, 8}
```