**AIM:-**

Write a Program to evaluate factorial , palindrome ,swapping two numbers and perform basic operations in lists, tuples, sets and dictionaries.

**PROGRAM:-**

a=int(input("enter first number:-")) b=int(input("enter second number:-")) print("numbers before swapping:-") print("a=",a,"b=",b)

swap=a a=b b=swap

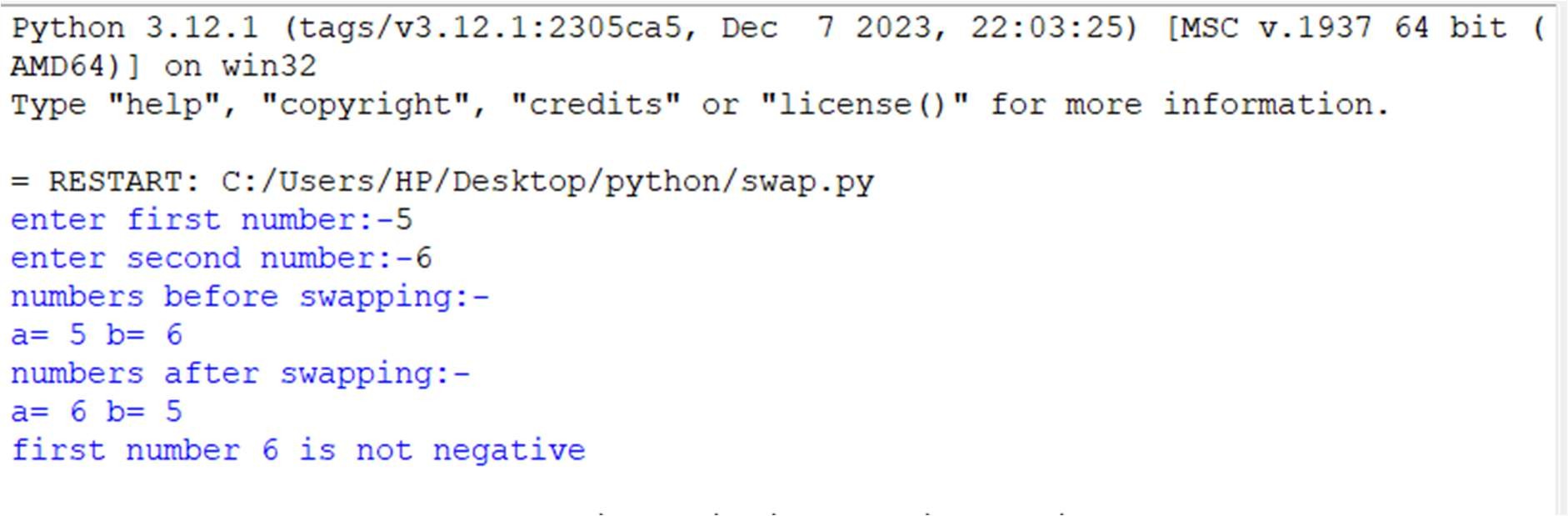
print("numbers after swapping:-") print("a=",a,"b=",b)

if a<0:

print("first number",a,"is negative") else:

print("first number",a,"is not negative")

**OUTPUT:-**



**THEORY:-**

A palindrome is a sequence of characters that reads the same forwards as backward. In the context of numbers, a palindrome number remains unchanged when its digits are reversed. Palindromes are interesting in various fields, including mathematics and computer science.The factorial of a non-negative integer is the product of all positive integers less than or equal to that number. It is denoted by the symbol !. The factorial function is fundamental in combinatorics and probability theory.

**PROGRAM:-**

1. **string Palindrome or not** str=input(("enter the string:-")) print("your entered string is",str) str1=""

for i in range(0,len(str)):

a=str[i] str1=a+str1

if str==str1:

print("given string",str,"is palindrome") else:

print("given string",str,"is not a palindrome")

# B]Factorial of given number

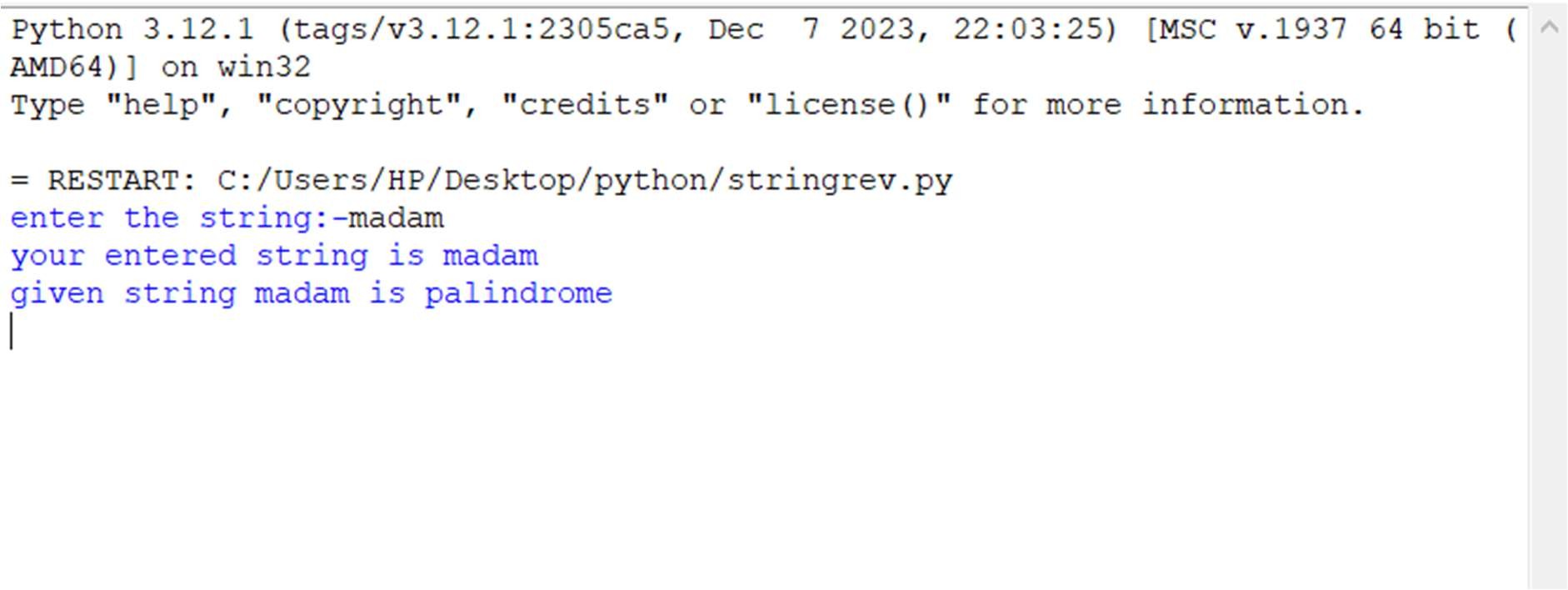
fact=int(input("enter the number whose factorial is to be found:-")) a=fact

for i in range(1,fact): fact=fact\*i

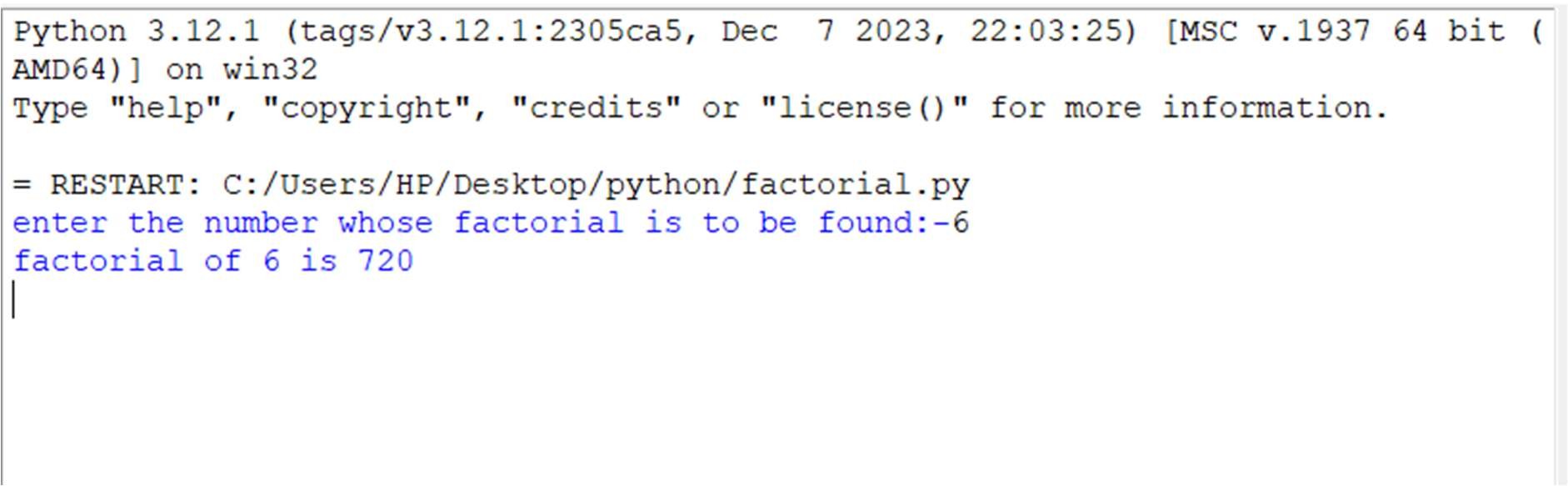
print("factorial of",a,"is",fact)

**OUTPUT:-**

**A]**



**B]**



**THEORY:-**

Python's list is a flexible, versatile, powerful, and popular built-in data type. It allows you to create variable-length and mutable sequences of objects. In a list, you can store objects of any type. You can also mix objects of different types within the same list, although list elements often share the same type.

**PROGRAM:-**

mylist=[]

n=int(input("enter the size of list:-")) print("enter the elementss of list:-") for i in range(0,n):

ele=int(input()) mylist.append(ele)

print("the list is:-",mylist)

print("1.create list of odd elements\n2.create list of even elements\n3.merge the two lists\n4.sort the two lists\n5.Update first element of list with X value\n6.delete the middle element of the list\n7.Find the min and max froM

the list\n8. Add N names into the existing number list and check if whether python name is present or not\n9.press 9 for exit")

while True:

x=int(input("enter your choice:-")) if x==1:

oddlist=[i for i in mylist if i%2!=0] print(oddlist)

elif x==2:

evenlist=[i for i in mylist if i%2==0] print(evenlist)

elif x==3: list1=[] list2=[] list3=[]

lim1=int(input("enter the limit of the first list:-")) print("enter the elements of first list:-")

for i in range(0,lim1): ele=int(input()) list1.append(ele)

lim2=int(input("enter the limit of second list:-")) print("enter the elements of the second list:-") for i in range(0,lim2):

ele=int(input()) list2.append(ele)

list3=list1+list2

print("the merged list is:-",list3)

elif x==4: list1=[] list2=[] list3=[]

lim1=int(input("enter the limit of the first list:-")) print("enter the elements of first list:-")

for i in range(0,lim1): ele=int(input()) list1.append(ele)

lim2=int(input("enter the limit of second list:-")) print("enter the elements of second list:-")

for i in range(0,lim2): ele=int(input()) list2.append(ele)

list3=list1+list2 list3.sort()

print("the sorted list is:-",list3) elif x==5:

a=int(input("enter value which is to be replaced with first element of list:-

"))

print("list before replacing:-",mylist) mylist[0]=a

print("list after replacing:-",mylist)

elif x==6:

if len(mylist)%2==0: mid=len(mylist)//2

else:

mid=len(mylist)//2

print("list before deleting middle element:-",mylist) mylist.pop(mid)

print("list after deleting middle element:-",mylist) elif x==7:

minimum=min(mylist) maximum=max(mylist)

print("minimum element of list is:-",minimum,"\nmax element of list:- ",maximum)

elif x==8:

lim=int(input("enter the number of elements to be inserted:-")) for i in range(0,lim):

ele=input() mylist.append(ele)

for i in range(0,len(mylist)): if mylist[i]=='python':

ﬂag=1 else:

ﬂag=0 if ﬂag==1:

print("python is present in the list") else:

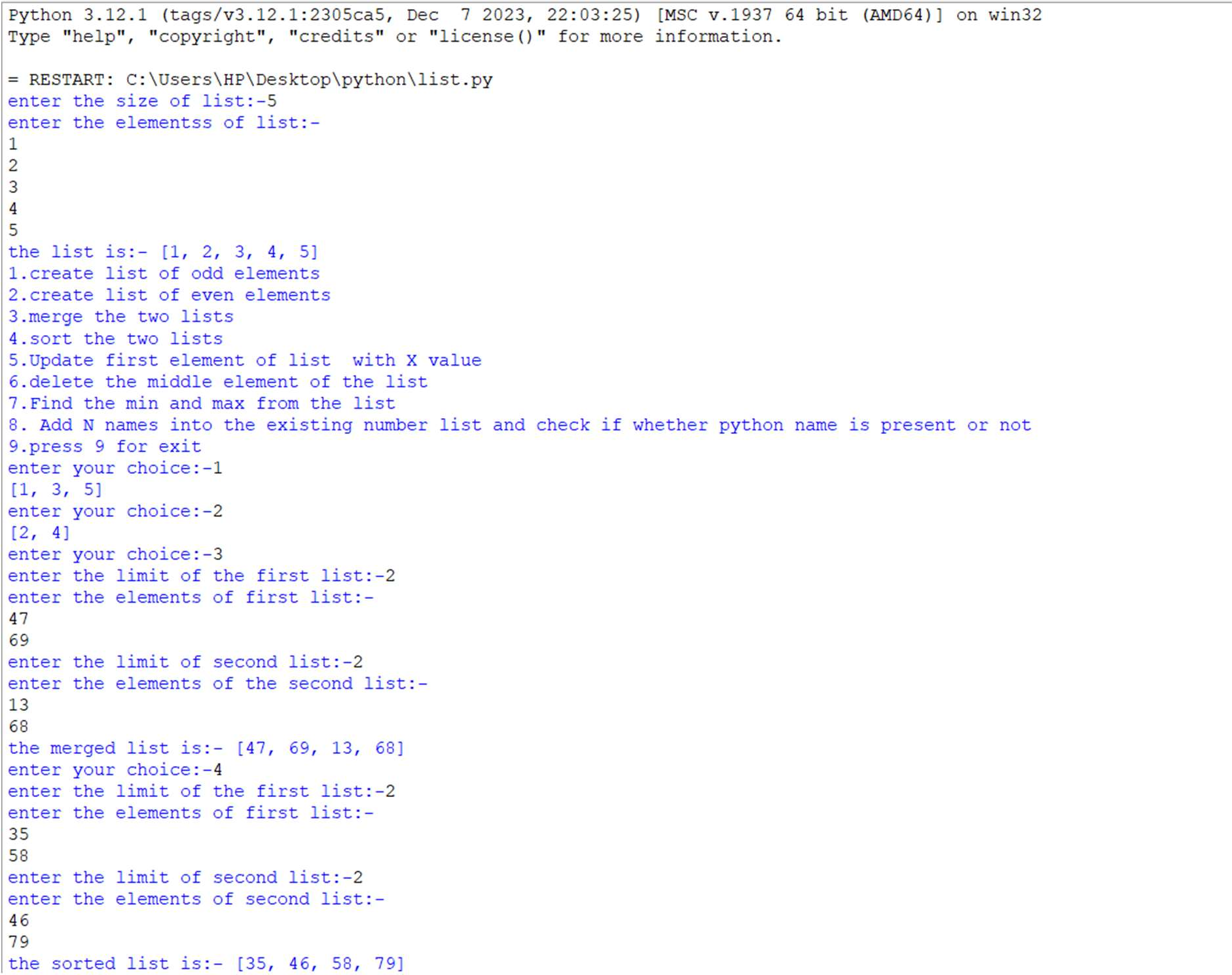
print("python is not present in the list") elif x==9:

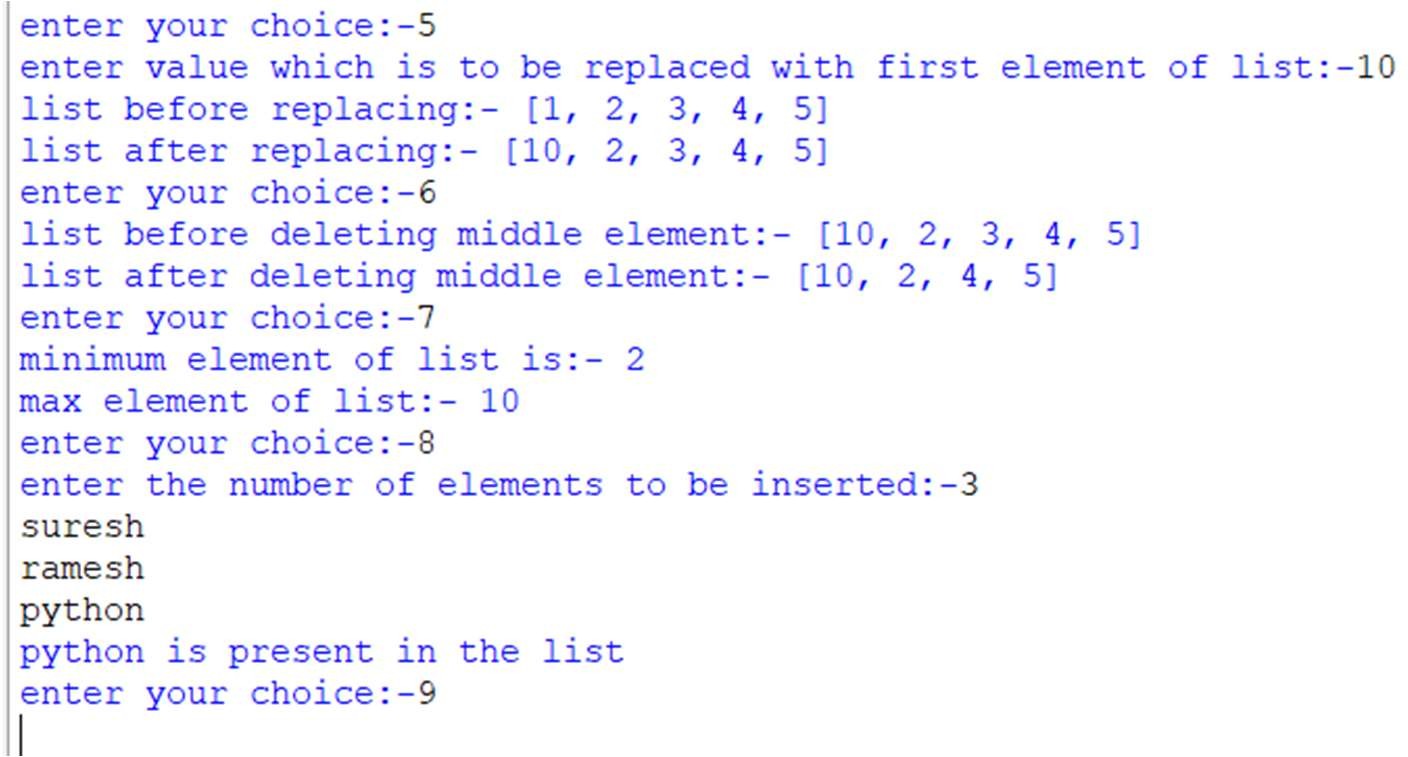
break

else:

print("invalid option enter valid option")

**OUTPUT:-**





**THEORY:-**

Python tuples are a type of data structure that is very similar to lists. The main difference between the two is that tuples are immutable, meaning they cannot be changed once they are created. This makes them ideal for storing data that should not be modified, such as database records.

**PROGRAM:-**

names=()

roll\_number=()

maths=()

science=()

english=()

print("1.To add information of N nuber of students(Name,Roll number,3 subject marks)\n\n2.To display student's roll number and marks whose name is 'python'\n\n3.To perform nested tuples and Sort nested tuple by name.\n\n4.press(4)for exit")

while True:

x=int(input("enter your choice:-"))

if x==1:

lim=int(input("enter the number of students:-"))

print("enter the name of students:-")

for i in range(0,lim):

ele=input()

names=names+(ele,)

print("enter roll number of students:-")

for i in range(0,lim):

ele=int(input())

roll\_number=roll\_number+(ele,)

print("enter the marks of maths of",lim,"students")

for i in range(0,lim):

ele=int(input())

maths=maths+(ele,)

print("enter the marks of science of",lim,"students")

for i in range(0,lim):

ele=int(input())

science=science+(ele,)

print("enter the marks of science of",lim,"students")

for i in range(0,lim):

ele=int(input())

english=english+(ele,)

print("NAME OF THE STUDENTS:-",names)

print("ROLL NUMBER OF STUDENTS:-",roll\_number)

print("MATHEMATICS MARKS OF ABOVE STUDENTS:-",maths)

print("SCIENCE MARKS OF ABOVE STUDENTS:-",science)

print("ENGLISH MARKS OF ABOVE STUDENTS:-",english)

elif x==2:

student\_names=()

student\_number=()

student\_marks=()

lim=int(input("enter the number of students:-"))

print("enter the name of students:-")

for i in range(0,lim):

ele=input()

student\_names=student\_names+(ele,)

print("enter the roll number of students:-")

for i in range(0,lim):

ele=int(input())

student\_number=student\_number+(ele,)

print("enter total marks of students:-")

for i in range(0,lim):

ele=int(input())

student\_marks=student\_marks+(ele,)

for i in range(0,lim):

if student\_names[i]=="python":

index=i

flag=1

break

else:

flag=0

if flag==1:

print("Student with name 'python' is present")

print("Roll Number is:-",student\_number[index])

print("total marks are:-",student\_marks[index])

else:

print("Student with name 'python' is not present")

elif x==3:

nested\_tupple=()

tuple1=()

lim1=int(input("enter the size of tupple:-"))

for i in range(0,lim1):

fruit\_name=input("enter fruit names:-")

fruit\_colour=input("enter colour of fruits:-")

fruit\_quant=int(input("enter quantity of fruits:-"))

tuple1=(fruit\_name,fruit\_colour,fruit\_quant)

nested\_tupple=nested\_tupple+(tuple1,)

print("tupple before sorting:-",nested\_tupple)

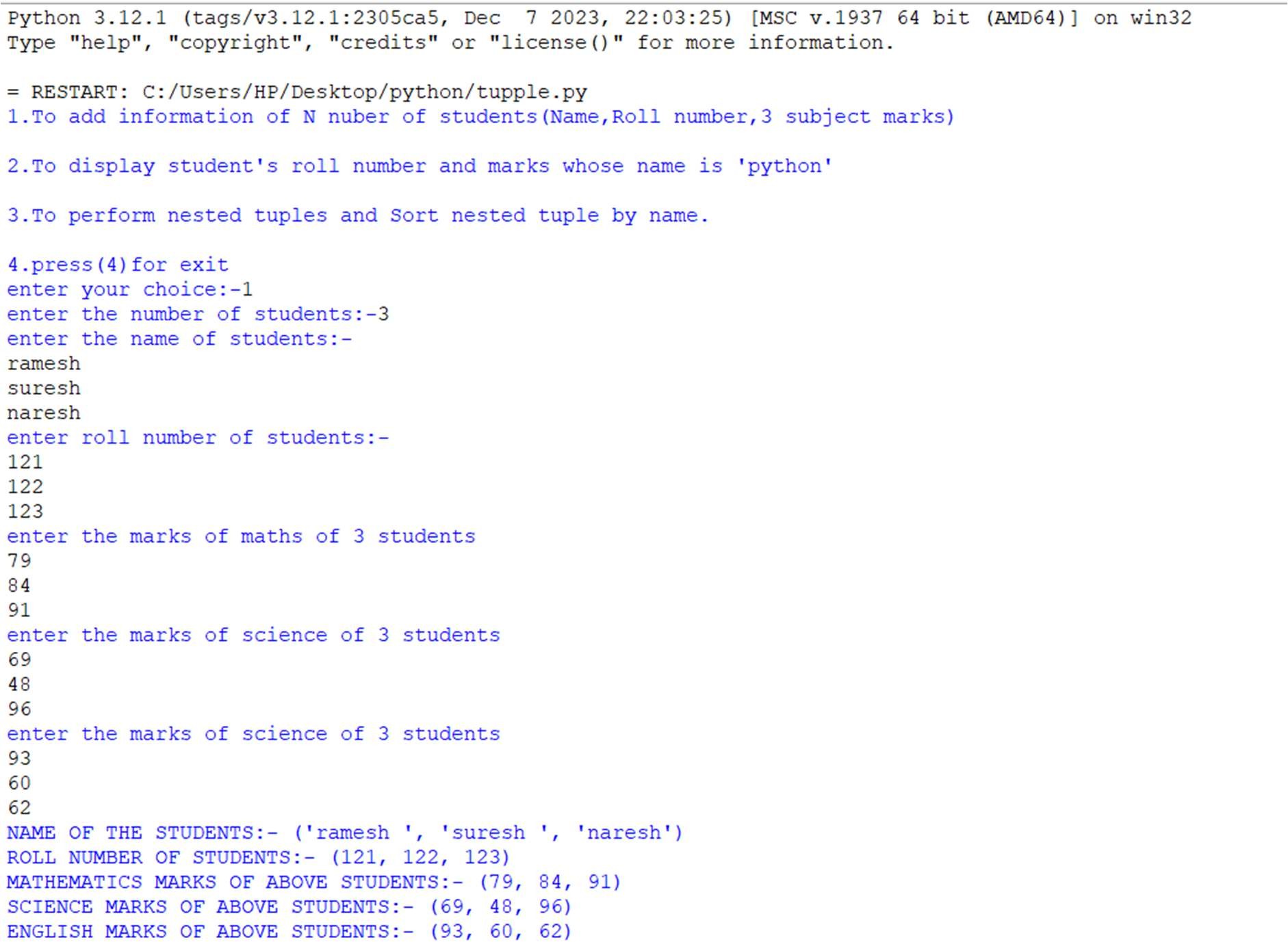
print("tupple after sorting:-",sorted(nested\_tupple))

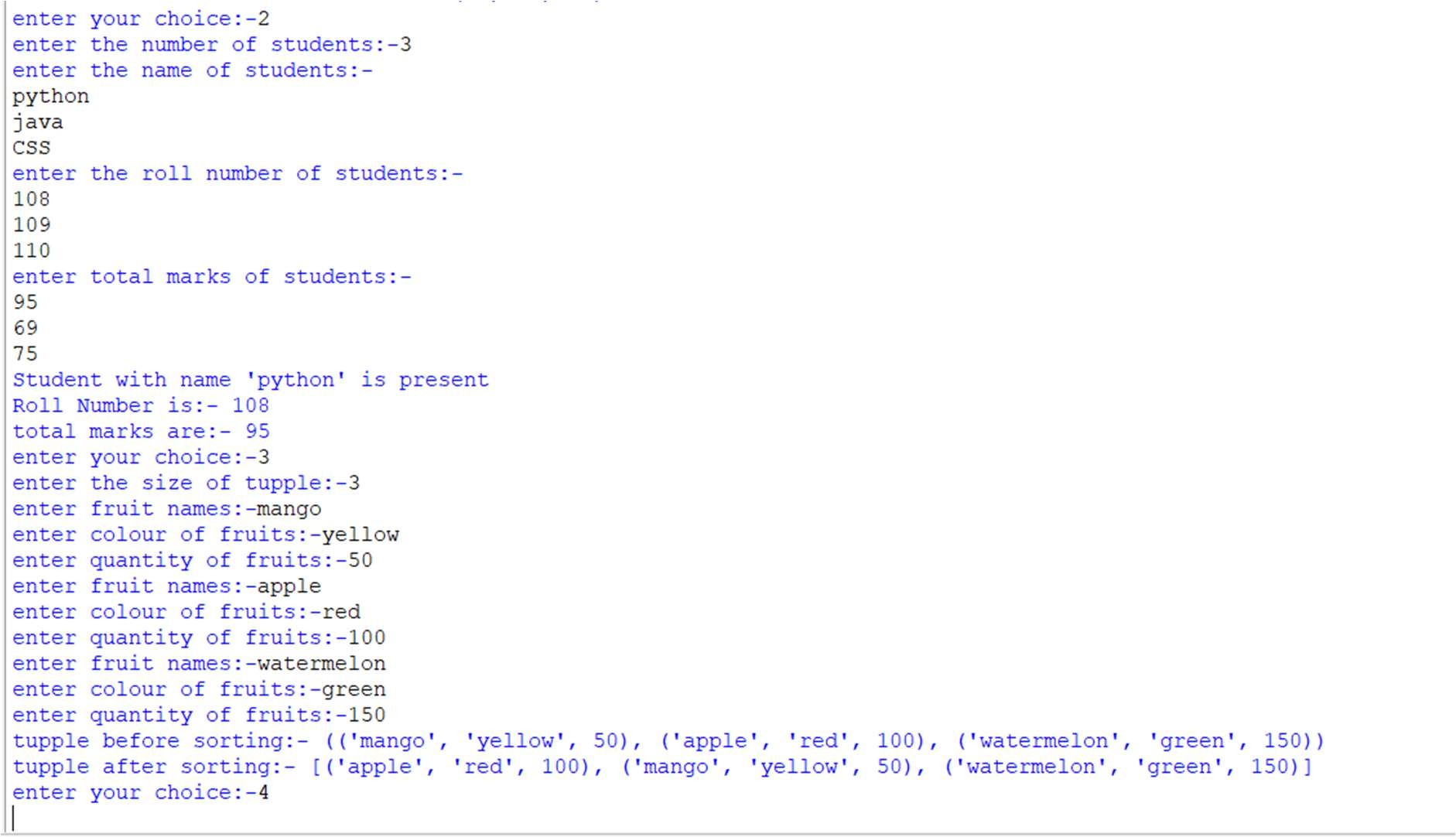
elif x==4:

break

else:

print("invalid option enter valid option")

**OUTPUT:-**



**THEORY:-**

* Sets are unordered.
* Set elements are unique. Duplicate elements are not allowed.
* A set itself may be modified, but the elements contained in the set must be of an [immutable](https://realpython.com/python-mutable-vs-immutable-types/) type.

**PROGRAM:-**

import sys

class Experiment:

def stringinput(self):

str1 = str(input("Enter the first string "))

str2 = str(input("Enter the second string "))

print("Strings after getting converted into set : ")

str1\_set = set(str1)

str2\_set = set(str2)

print("The set string 1 is :" ,str1\_set)

print("The set string 2 is : " ,str2\_set)

def set\_union(self):

str1 = str(input("Enter the first string "))

str2 = str(input("Enter the second string "))

print("Strings after getting converted into set : ")

str1\_set = set(str1)

str2\_set = set(str2)

print("The set string 1 is :" ,str1\_set)

print("The set string 2 is : " ,str2\_set)

print("The two set union is " , str1\_set | str2\_set )

def set\_intersection(self):

str1 = str(input("Enter the first string "))

str2 = str(input("Enter the second string "))

print("Strings after getting converted into set : ")

str1\_set = set(str1)

str2\_set = set(str2)

print("The set string 1 is :" ,str1\_set)

print("The set string 2 is : " ,str2\_set)

print("The two set intersection is " , str1\_set & str2\_set )

def set\_differnce(self):

str1 = str(input("Enter the first string "))

str2 = str(input("Enter the second string "))

print("Strings after getting converted into set : ")

str1\_set = set(str1)

str2\_set = set(str2)

print("The set string 1 is :" ,str1\_set)

print("The set string 2 is : " ,str2\_set)

print("The difference between set1 from set2 " , str1\_set - str2\_set )

def set\_symmetricdiffernce(self):

str1 = str(input("Enter the first string "))

str2 = str(input("Enter the second string "))

print("Strings after getting converted into set : ")

str1\_set = set(str1)

str2\_set = set(str2)

print("The set string 1 is :" ,str1\_set)

print("The set string 2 is : " ,str2\_set)

print("The two set's symmetric difference is " , str1\_set ^ str2\_set )

obj1 = Experiment()

def call():

while True:

print("Press 1 : For Creating two strings ")

print("Press 2 : For Displaying Set Union of two strings")

print("Press 3 : For displaying Set intersection of two strings")

print("Press 4 : For Illustrating Set difference between two strings")

print("Press 5 : For dispalying symmetric difference of two strings")

print("Press 6 : For exiting")

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

if choice == 1 :

obj1.stringinput()

elif choice == 2:

obj1.set\_union()

elif choice == 3 :

obj1.set\_intersection()

elif choice == 4 :

obj1.set\_differnce()

elif choice == 5 :

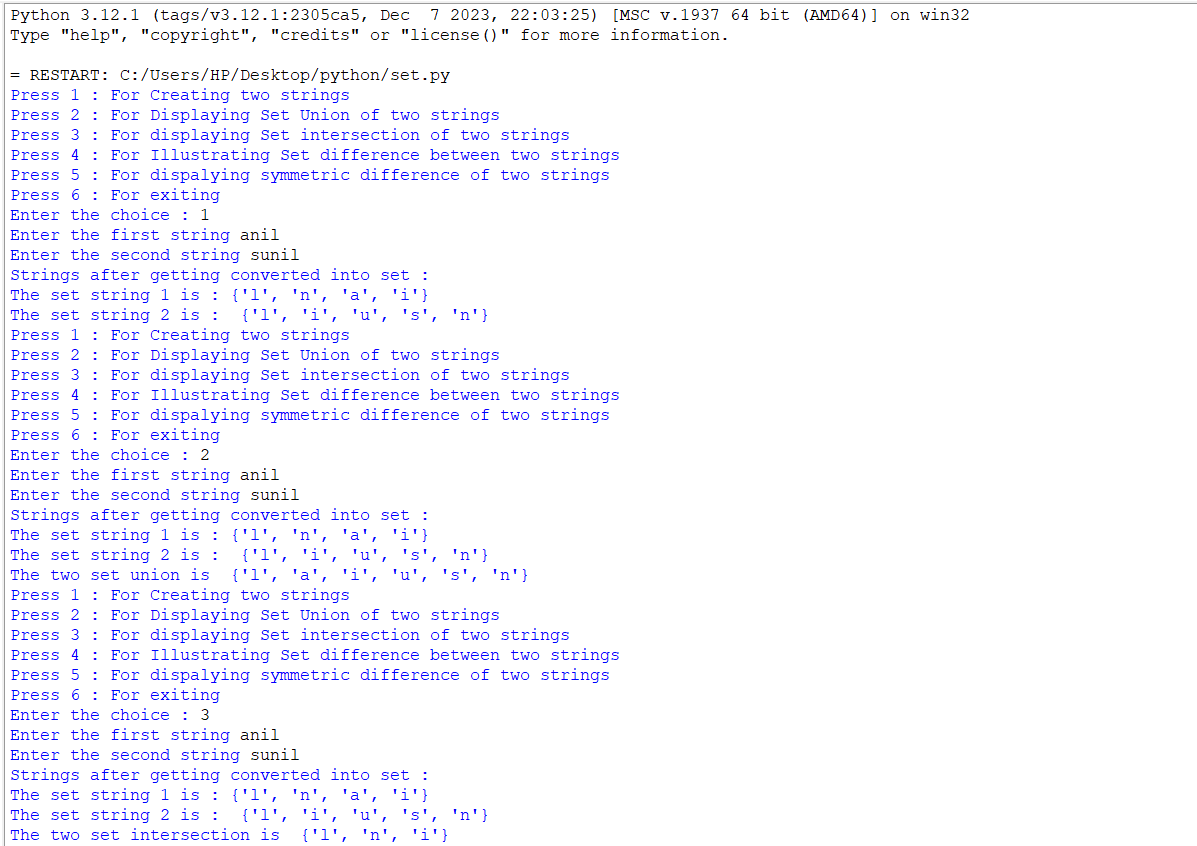
obj1.set\_symmetricdiffernce()

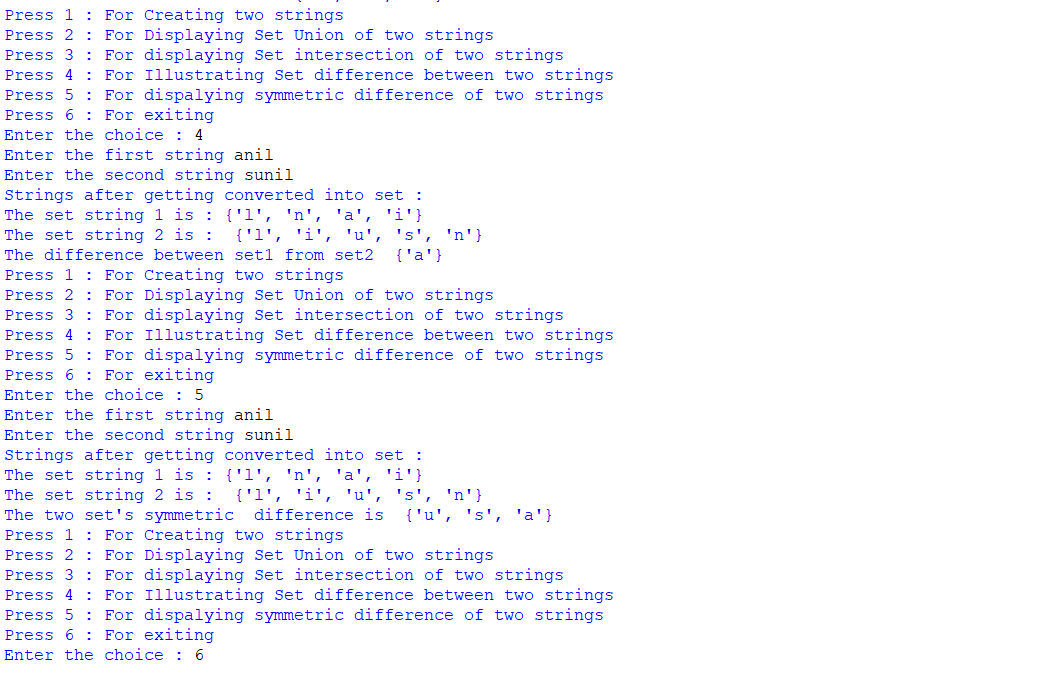
elif choice == 6:

sys.exit()

call()

**OUTPUT:-**





**THEORY:-**

Dictionaries and lists share the following characteristics:

* Both are mutable.
* Both are dynamic. They can grow and shrink as needed.
* Both can be nested. A list can contain another list. A dictionary can contain another dictionary. A dictionary can also contain a list, and vice versa.

Dictionaries differ from lists primarily in how elements are accessed:

* List elements are accessed by their position in the list, via indexing.
* Dictionary elements are accessed via keys.

**PROGRAM:-**

import sys

class Experiment:

def create\_dictionary(self):

dictionary = {}

lim = int(input("Enter the number of key:value pairs"))

for i in range(0,lim):

key=input("Enter the key Name ")

value=input("Enter the value of the key ")

dictionary[key]=value

print("The dictionary is :\n" , dictionary)

def exercise\_on\_dictionary(self):

dictionary = {}

lim = int(input("Enter the number of key:value pairs"))

for i in range(0,lim):

key=input("Enter the key ")

value=input("Enter the value of the key ")

dictionary[key]=value

print("The dictionary is :\n" , dictionary)

print("Concatenating another Key:Value pair")

key = input("Enter the key that is to be added :")

value =input("Enter the value of the key ")

dictionary[key]=value

print("The dictionary is :\n" , dictionary)

print("Deleting a particular Key:Value pair ")

key = input("Enter the key of the key:value pair that is to be deleted : ")

dictionary.pop(key)

print("The dictionary after deleting the key:valure pair " ,dictionary)

def traversing\_dictionary(self):

dictionary = {}

lim = int(input("Enter the number of key:value pairs"))

for i in range(0,lim):

key=input("Enter the key ")

value=input("Enter the value of the key ")

dictionary[key]=value

print("The dictionary is :\n" , dictionary)

found\_value = input("Enter the value that is to be found : ")

for i in dictionary:

if dictionary[i] == found\_value :

print("The value is found ")

elif(i == lim-1 and dictionary[i] != found\_value ):

print("Element not found ")

def maplist\_dictionary(self):

dictionary = {}

lim = int(input("Enter the number of key-value pairs in the dictionary: "))

for i in range(lim):

key = input("Enter the key: ")

value = []

limlist = int(input("Enter the limit of list : "))

print("Enter the values of list : ")

for i in range(0,limlist):

ele=input()

value.append(ele)

dictionary[key] = value

print("The dictionary is : " , dictionary)

obj1 = Experiment()

def call():

while True:

print("Press 1 - For Creating a Dictionary ")

print("Press 2 - For Updating and Deleting aa key:value pair")

print("Press 3 - For finding a key:value pair ")

print("Press 4 - For mapping 'n' list in dictionary ")

print("Press 5 - For exiting ")

choice = int(input("Enter your choice"))

if choice == 1:

obj1.create\_dictionary()

elif choice == 2:

obj1.exercise\_on\_dictionary()

elif choice == 3:

obj1.traversing\_dictionary()

elif choice == 4:

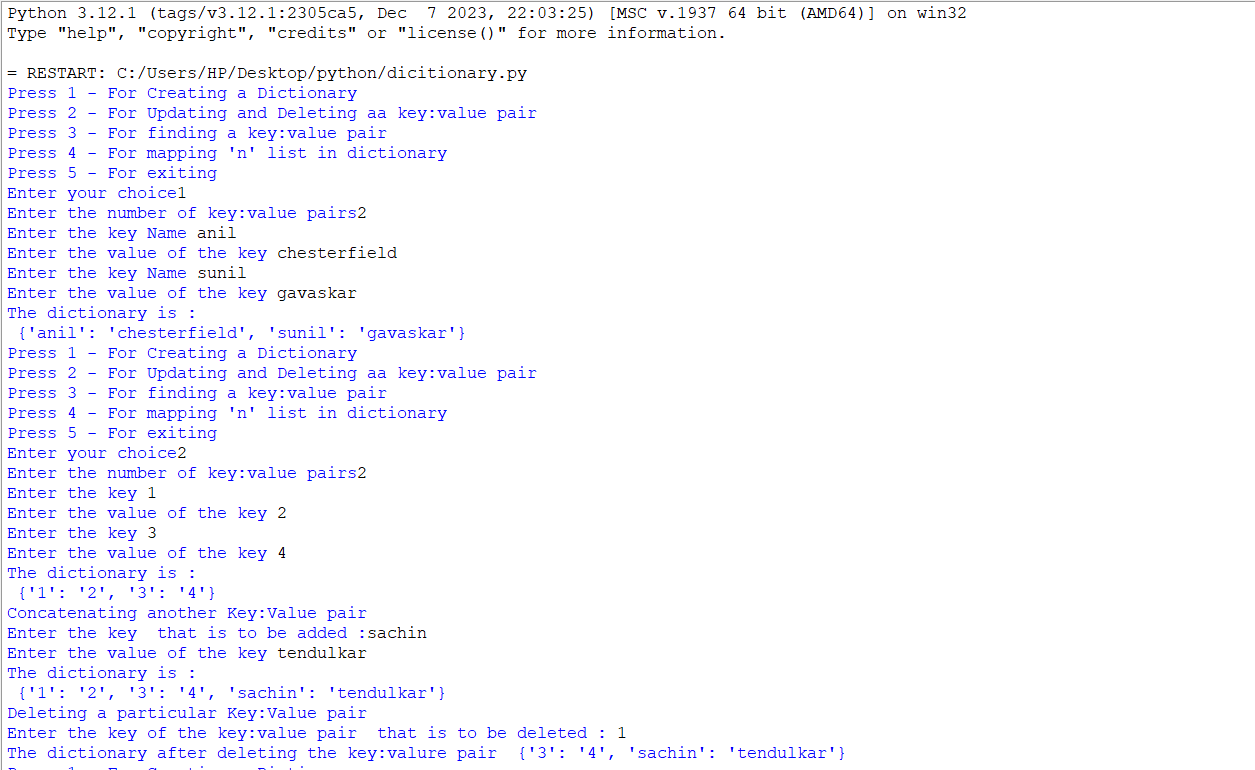
obj1.maplist\_dictionary()

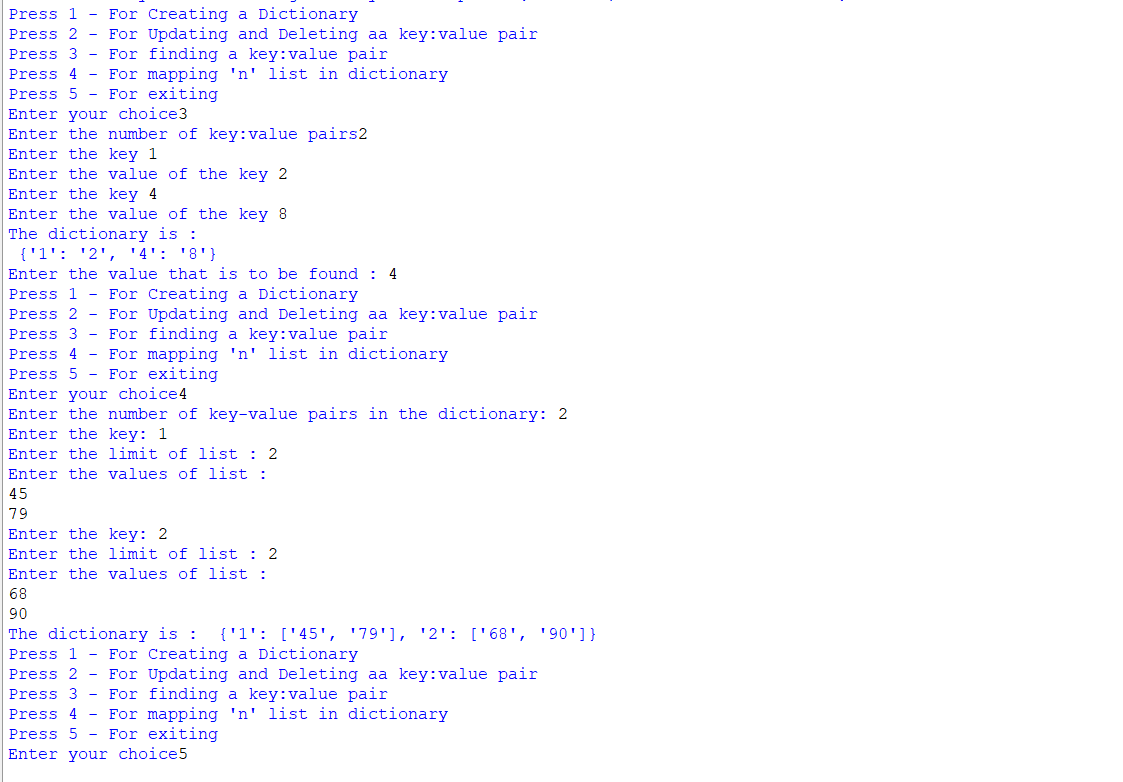
elif choice == 5:

sys.exit()

call()

**OUTPUT:-**





**CONCLUSION:-**

i).To acknowledge basic terminologies of Python .

ii).We were able to acknowledge existing Python Syntaxes by executing basic programs

iii).We were able to comprehensively denominate various list built-in data of Python characteristics

iv).We were able to contemplate built-in data type tuple in Python

v).We were able to summon various sets characteristics in python

vi).We were able to stipulate operations on built-in data type dictionary in Python .