MALAD KANDIVALI EDUCATION SOCIETY'S

NAGINDAS KHANDWALA COLLEGE OF COMMERCE, ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS KHANDWALA COLLEGE OF SCIENCE

MALAD [W], MUMBAI – 64

AUTONOMOUS INSTITUTION

(Affiliated To University Of Mumbai)

Reaccredited 'A' Grade by NAAC | ISO 9001:2015 Certified

<u>CERTIFICATE</u>

Name: Mr. __________ HIMANSHU SHARAD DUBEY ____________

Roll No: <u>**315**</u> **Programme:** BSc CS **Semester:** III

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **<u>Data Structures</u>** **<u>(Course Code : 2032UISPR)</u>** for the partial fulfilment of Third Semester of BSc IT during the academic year 2020-21.

The journal work is the original study work that has been duly approved in the year 2020-21 by the undersigned.

External Examiner

Mr. Gangashankar Singh
(Subject-In-Charge)

Date of Examination: **(College Stamp)**

Class: S.Y. B.Sc. CS Sem - III **Roll No: __315__**

Subject: Data Structures

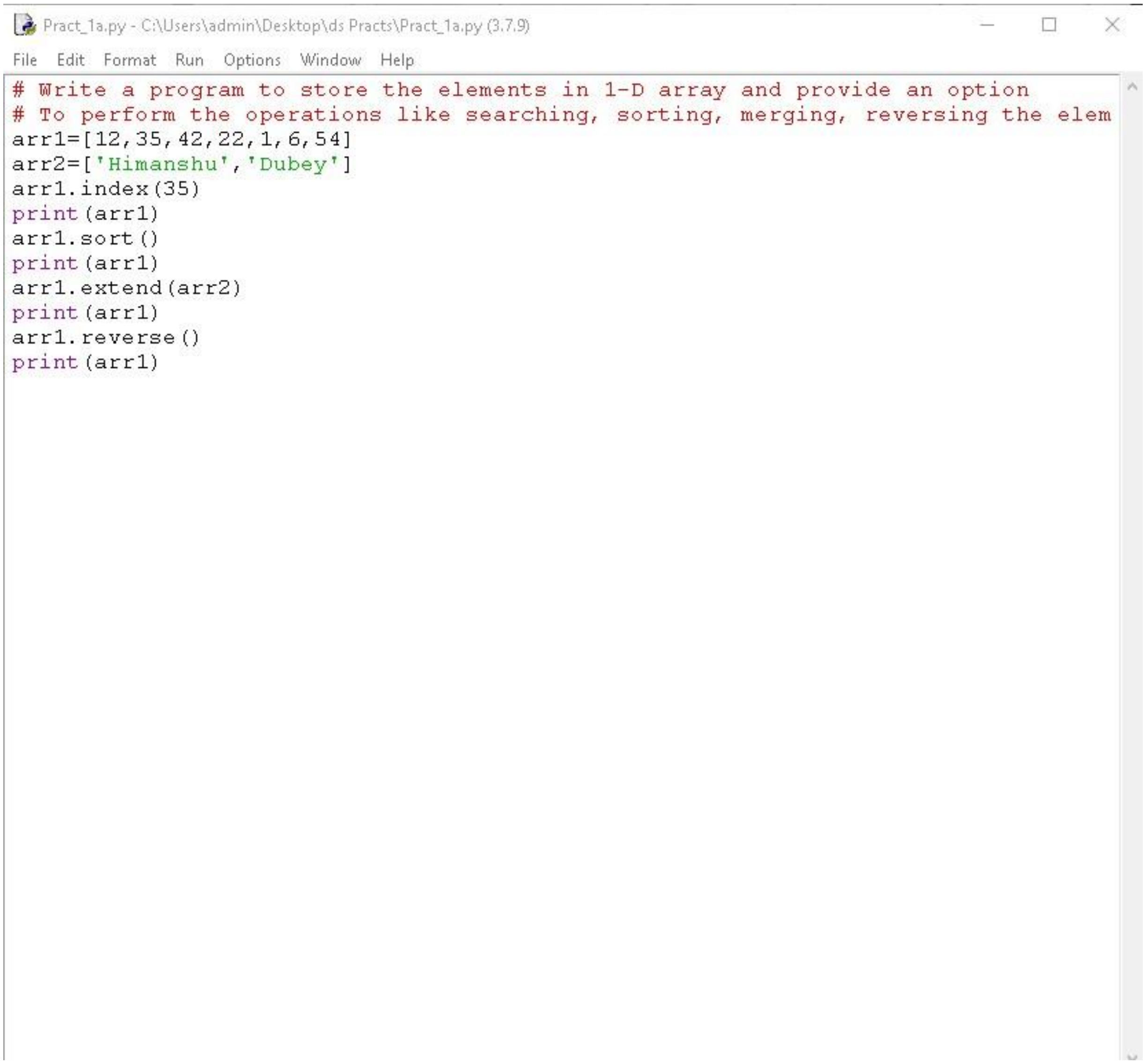
INDEX

PRACTICAL NO: -1A

AIM: - 1a) write a program to store the elements in 1_D array and provide an option to perform the operations like searching, sorting, merging, reversing the elements.

LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Ds%20pract_1a.py

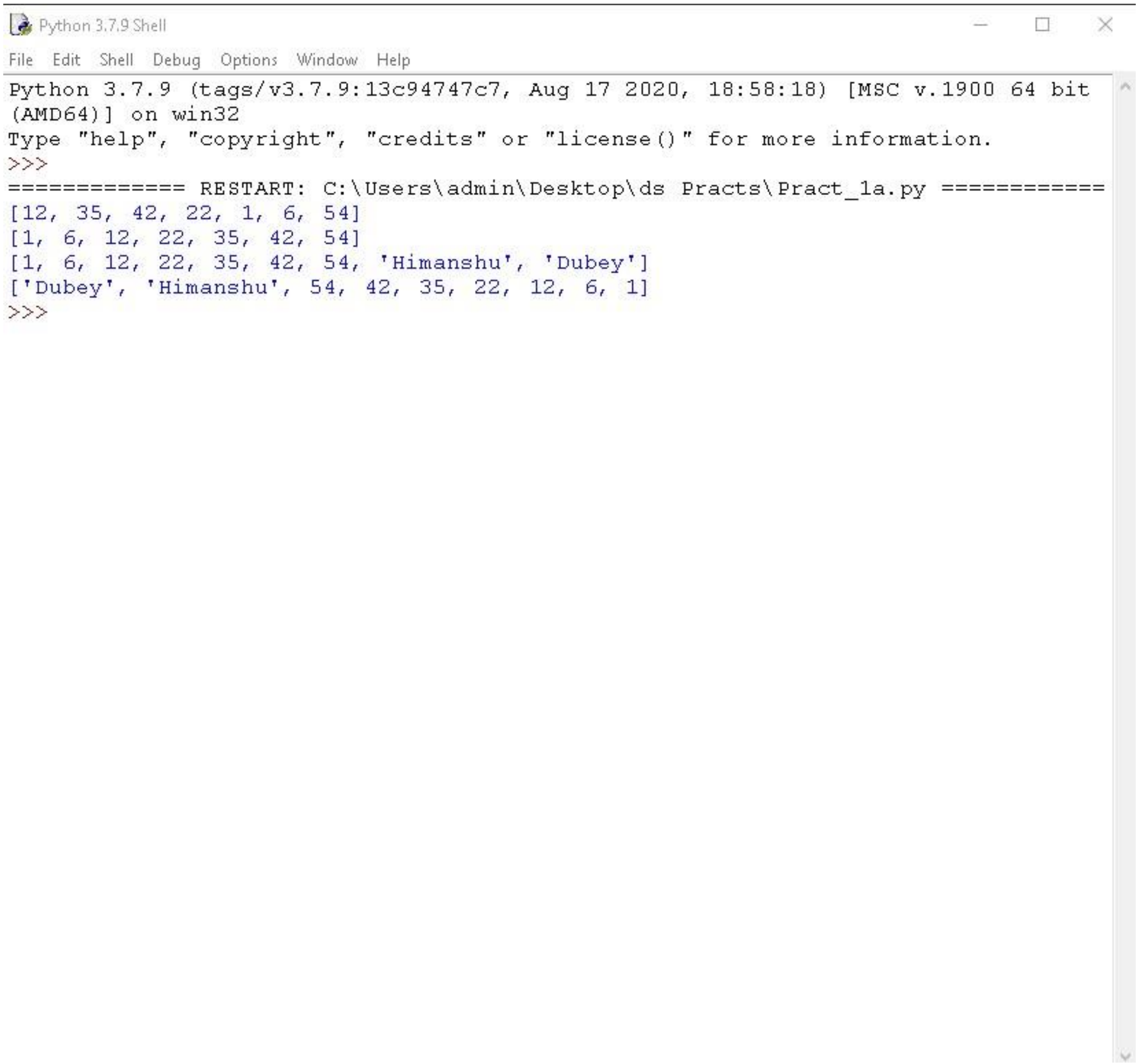
CODE:



Pract_1a.py - C:\Users\admin\Desktop\ds Practs\Pract_1a.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
# Write a program to store the elements in 1-D array and provide an option
# To perform the operations like searching, sorting, merging, reversing the elem
arr1=[12,35,42,22,1,6,54]
arr2=['Himanshu','Dubey']
arr1.index(35)
print(arr1)
arr1.sort()
print(arr1)
arr1.extend(arr2)
print(arr1)
arr1.reverse()
print(arr1)
```

OUTPUT:



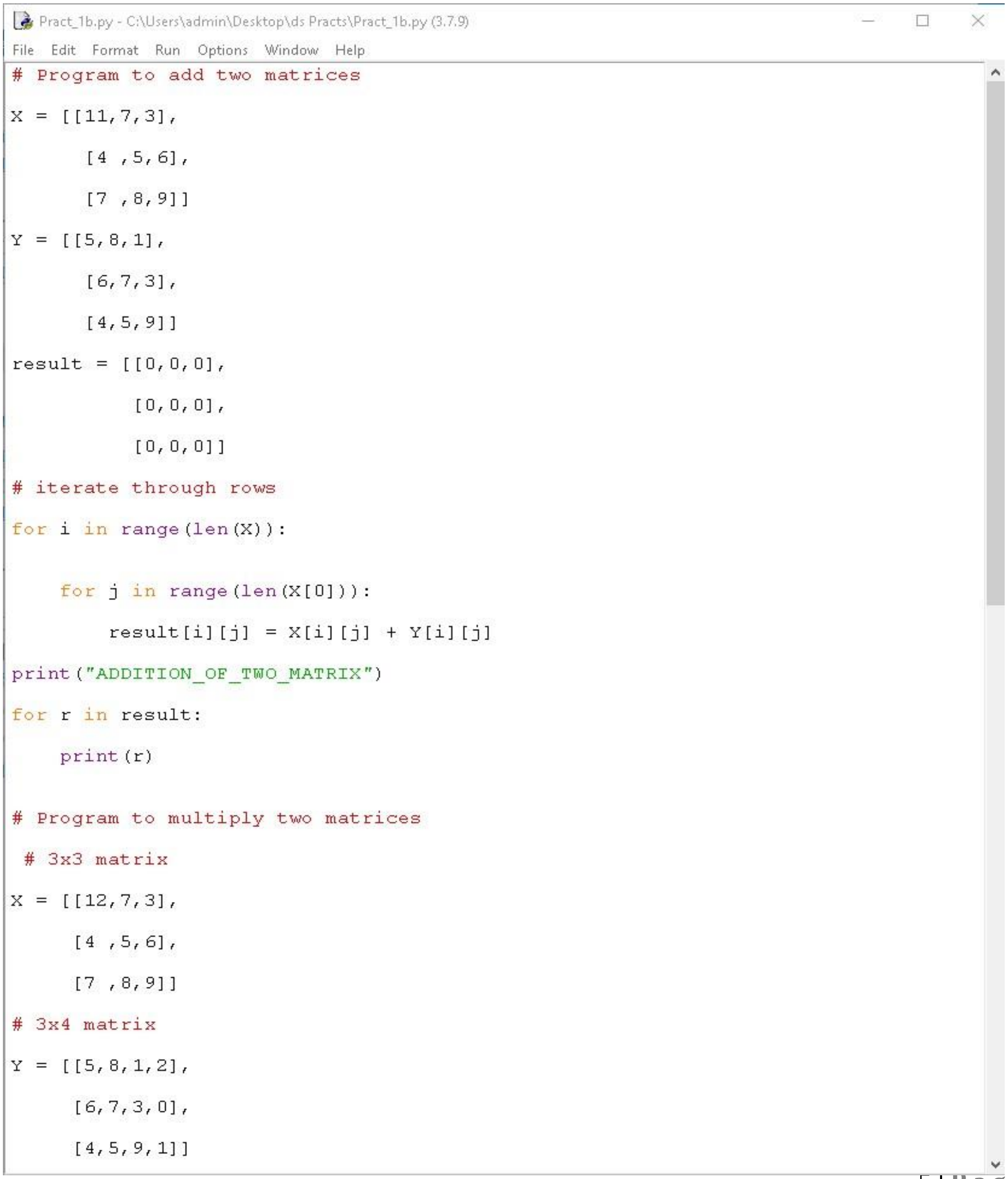
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_1a.py ===========
[12, 35, 42, 22, 1, 6, 54]
[1, 6, 12, 22, 35, 42, 54]
[1, 6, 12, 22, 35, 42, 54, 'Himanshu', 'Dubey']
['Dubey', 'Himanshu', 54, 42, 35, 22, 12, 6, 1]
>>>
```

PRACTICAL NO: -1B

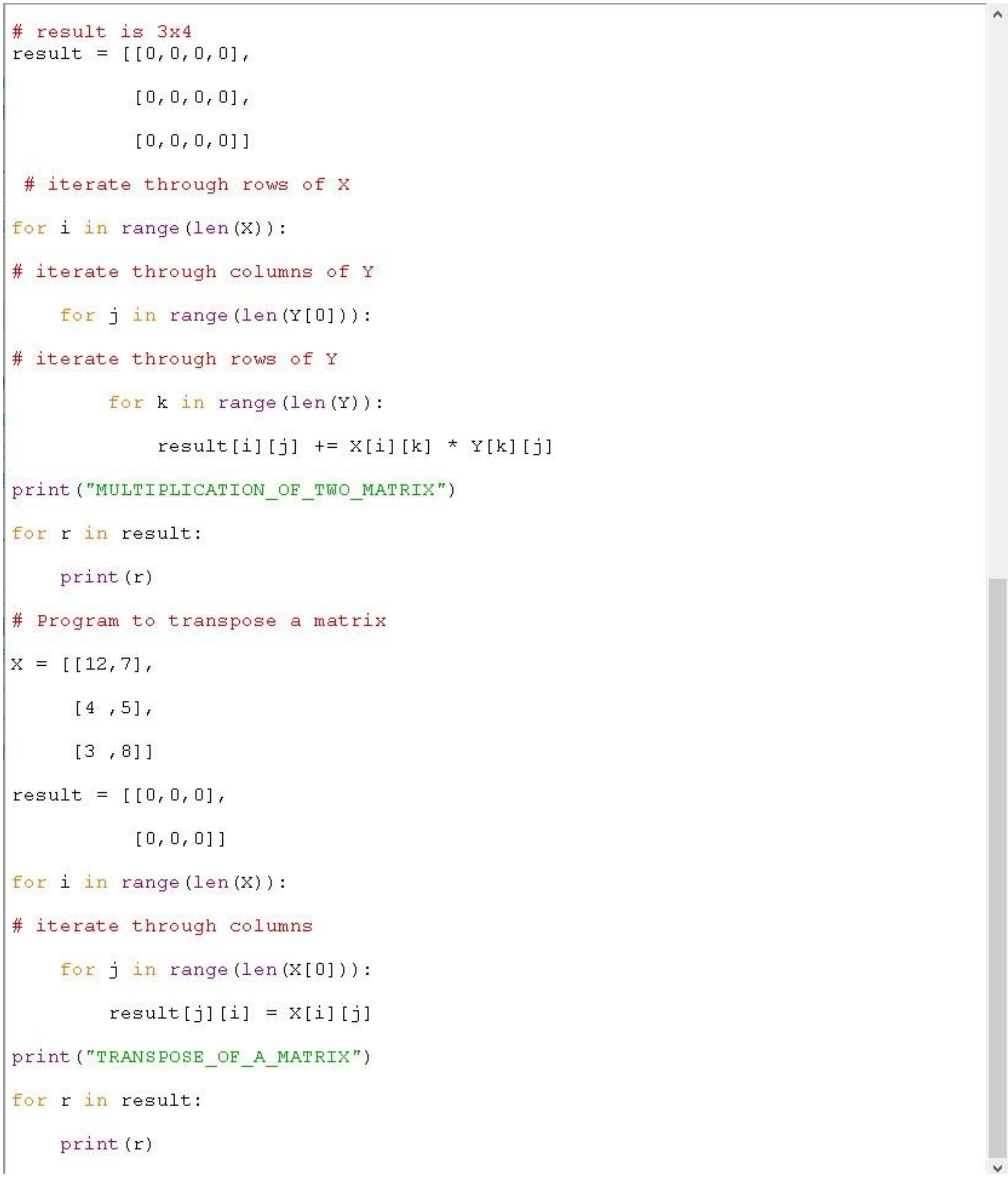
AIM: -1b) Write the program to perform the matrix addition, Multiplication and Transpose operation.

LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_1b.py

CODE:

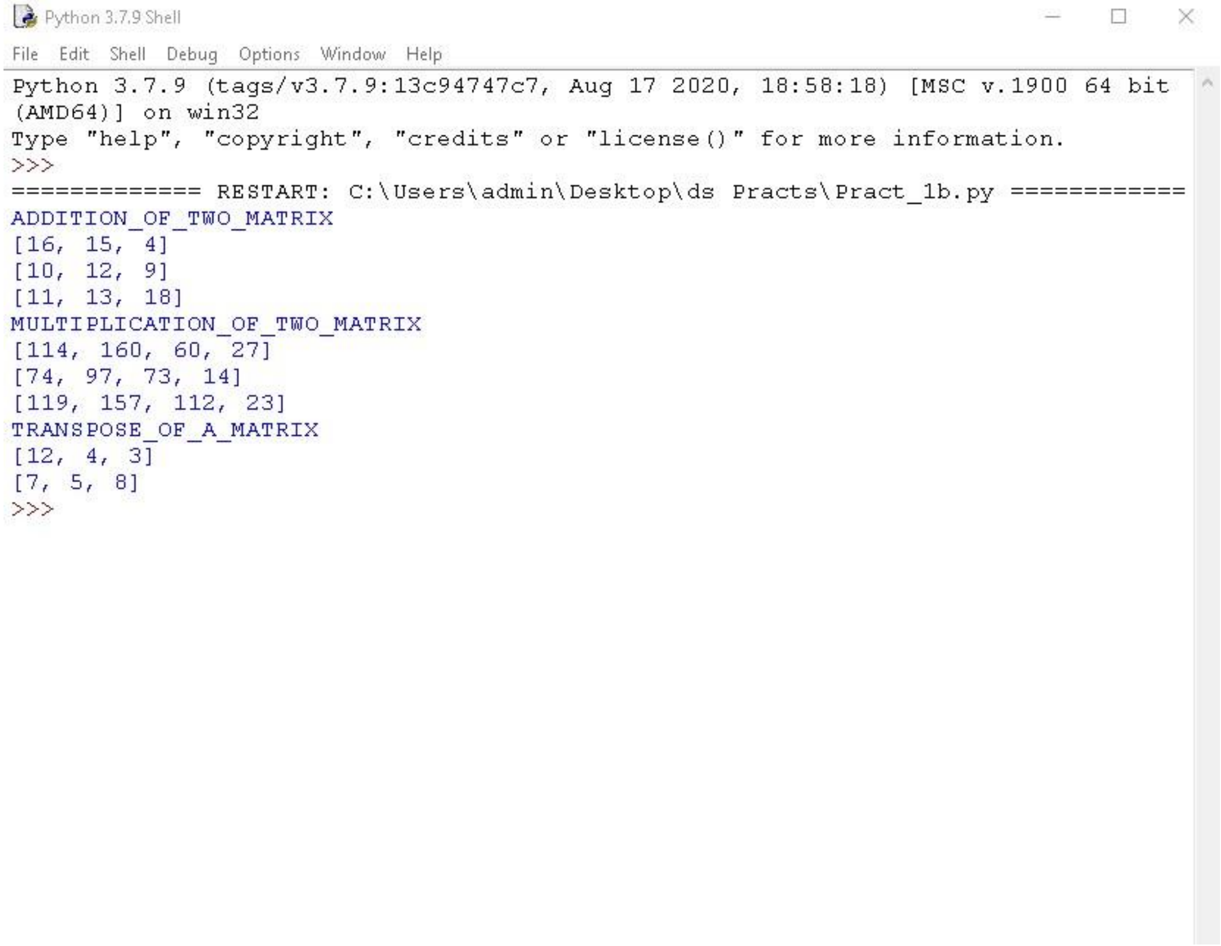

Pract_1b.py - C:\Users\admin\Desktop\ds Practs\Pract_1b.py (3.7.9)
File Edit Format Run Options Window Help

```
# Program to add two matrices
X = [[11,7,3],
      [4 ,5,6],
      [7 ,8,9]]
Y = [[5,8,1],
      [6,7,3],
      [4,5,9]]
result = [[0,0,0],
          [0,0,0],
          [0,0,0]]
# iterate through rows
for i in range(len(X)):

    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]
print("ADDITION_OF_TWO_MATRIX")
for r in result:
    print(r)

# Program to multiply two matrices
 # 3x3 matrix
X = [[12,7,3],
     [4 ,5,6],
     [7 ,8,9]]
# 3x4 matrix
Y = [[5,8,1,2],
     [6,7,3,0],
     [4,5,9,1]]
```



```
# result is 3x4
result = [[0,0,0,0],

          [0,0,0,0],

          [0,0,0,0]]

 # iterate through rows of X

for i in range(len(X)):

# iterate through columns of Y

    for j in range(len(Y[0])):

# iterate through rows of Y

        for k in range(len(Y)):

            result[i][j] += X[i][k] * Y[k][j]

print("MULTIPLICATION_OF_TWO_MATRIX")

for r in result:

    print(r)

# Program to transpose a matrix

X = [[12,7],

     [4 ,5],

     [3 ,8]]

result = [[0,0,0],

          [0,0,0]]

for i in range(len(X)):

# iterate through columns

    for j in range(len(X[0])):

        result[j][i] = X[i][j]

print("TRANSPOSE_OF_A_MATRIX")

for r in result:

    print(r)
```

OUTPUT:



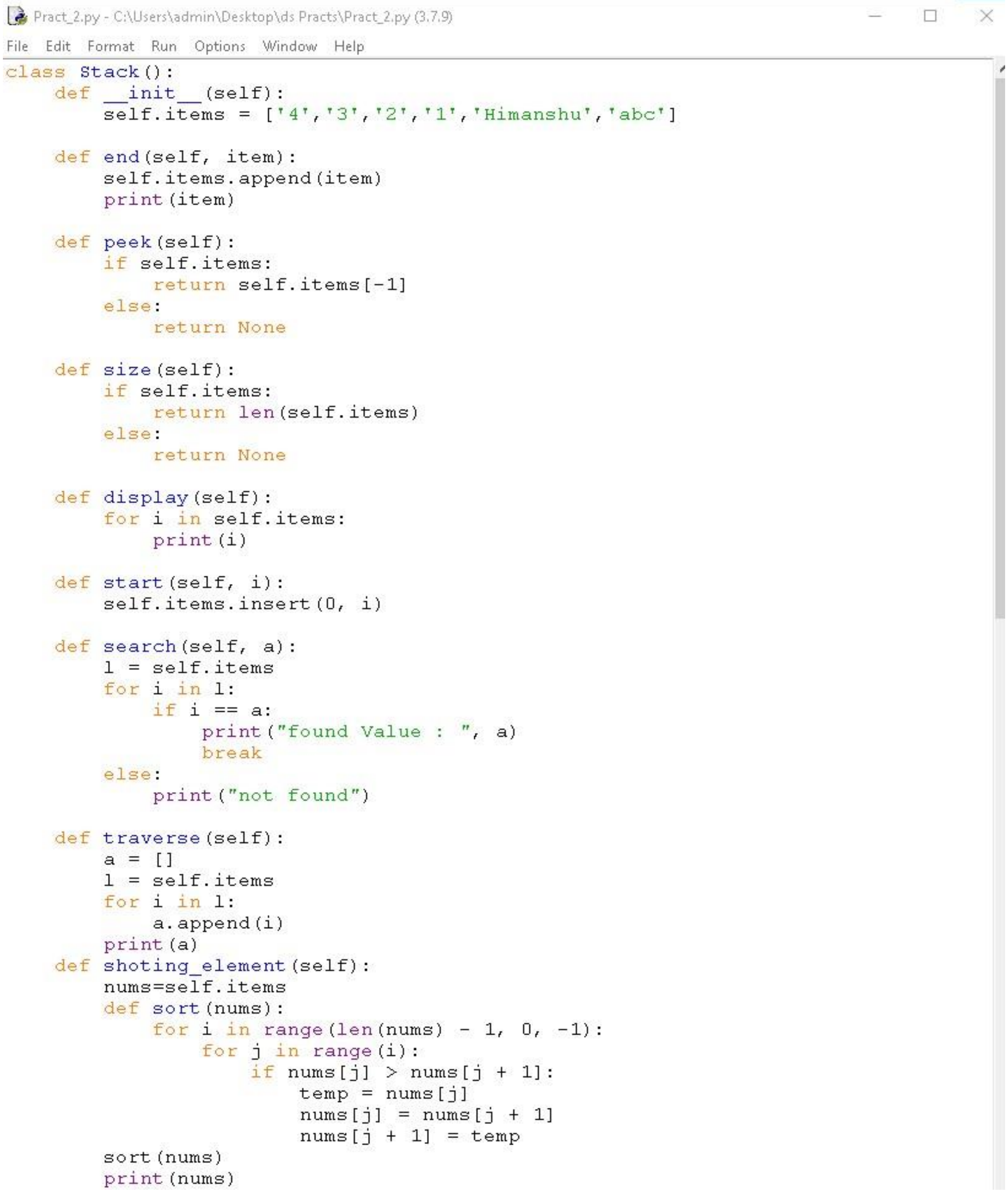
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_1b.py ============
ADDITION_OF_TWO_MATRIX
[16, 15, 4]
[10, 12, 9]
[11, 13, 18]
MULTIPLICATION_OF_TWO_MATRIX
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
TRANSPOSE_OF_A_MATRIX
[12, 4, 3]
[7, 5, 8]
>>>
```

PRACTICAL NO: -2

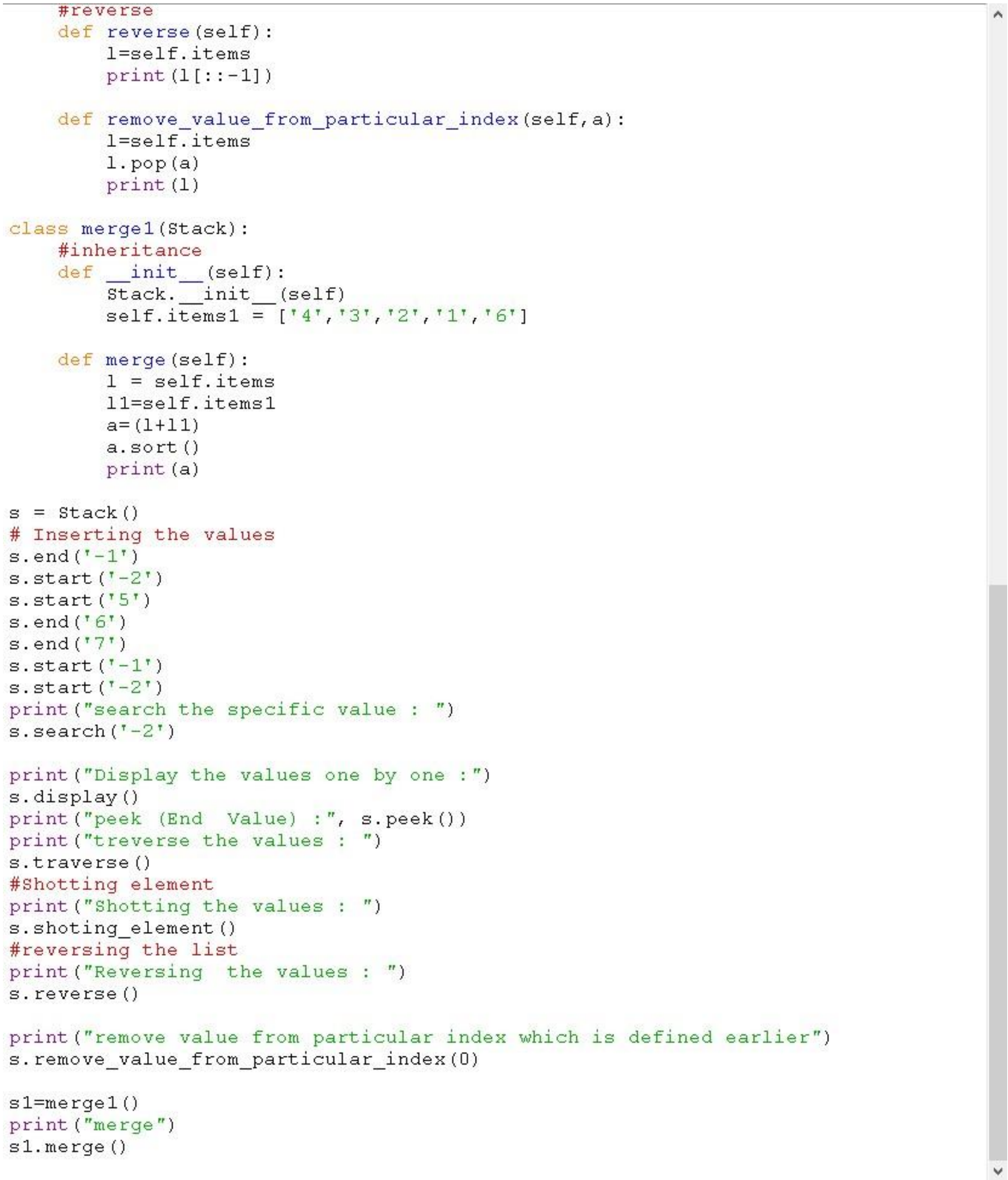
AIM: -2 Implement Linked List. Include options for Insertion, deletion and search of a number, reverse the list and concatenate the lists.

LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_2.py

CODE:

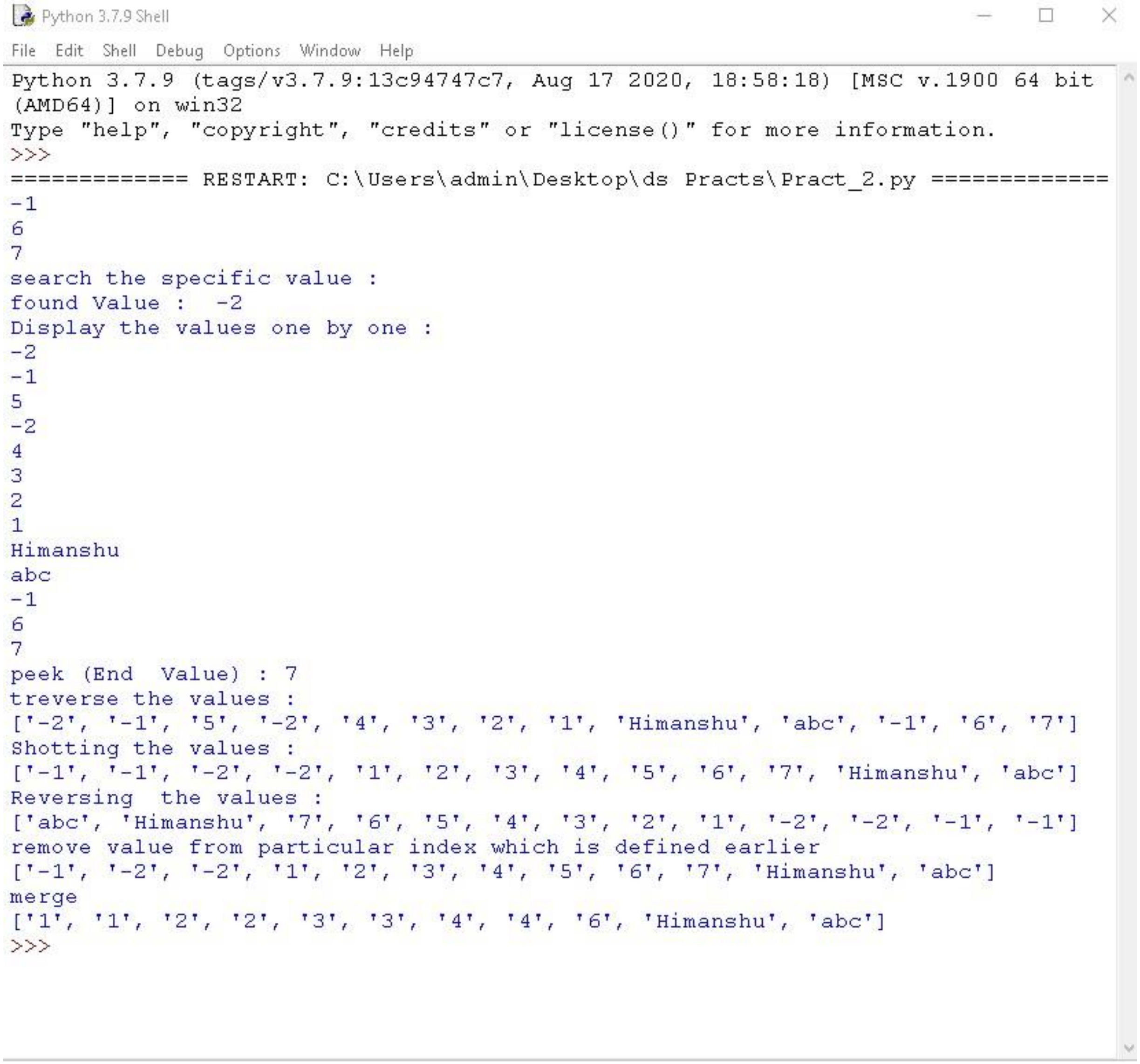

Pract_2.py - C:\Users\admin\Desktop\ds Practs\Pract_2.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
class Stack():
    def __init__(self):
        self.items = ['4','3','2','1','Himanshu','abc']

    def end(self, item):
        self.items.append(item)
        print(item)

    def peek(self):
        if self.items:
            return self.items[-1]
        else:
            return None

    def size(self):
        if self.items:
            return len(self.items)
        else:
            return None

    def display(self):
        for i in self.items:
            print(i)

    def start(self, i):
        self.items.insert(0, i)

    def search(self, a):
        l = self.items
        for i in l:
            if i == a:
                print("found Value : ", a)
                break
        else:
            print("not found")

    def traverse(self):
        a = []
        l = self.items
        for i in l:
            a.append(i)
        print(a)
    def shoting_element(self):
        nums=self.items
        def sort(nums):
            for i in range(len(nums) - 1, 0, -1):
                for j in range(i):
                    if nums[j] > nums[j + 1]:
                        temp = nums[j]
                        nums[j] = nums[j + 1]
                        nums[j + 1] = temp
        sort(nums)
        print(nums)
```



```
    #reverse
    def reverse(self):
        l=self.items
        print(l[::-1])

    def remove_value_from_particular_index(self,a):
        l=self.items
        l.pop(a)
        print(l)

class merge1(Stack):
    #inheritance
    def __init__(self):
        Stack.__init__(self)
        self.items1 = ['4','3','2','1','6']

    def merge(self):
        l = self.items
        l1=self.items1
        a=(l+l1)
        a.sort()
        print(a)

s = Stack()
# Inserting the values
s.end('-1')
s.start('-2')
s.start('5')
s.end('6')
s.end('7')
s.start('-1')
s.start('-2')
print("search the specific value : ")
s.search('-2')

print("Display the values one by one :")
s.display()
print("peek (End  Value) :", s.peek())
print("treverse the values : ")
s.traverse()
#Shotting element
print("Shotting the values : ")
s.shoting_element()
#reversing the list
print("Reversing  the values : ")
s.reverse()

print("remove value from particular index which is defined earlier")
s.remove_value_from_particular_index(0)

s1=merge1()
print("merge")
s1.merge()
```

OUTPUT:



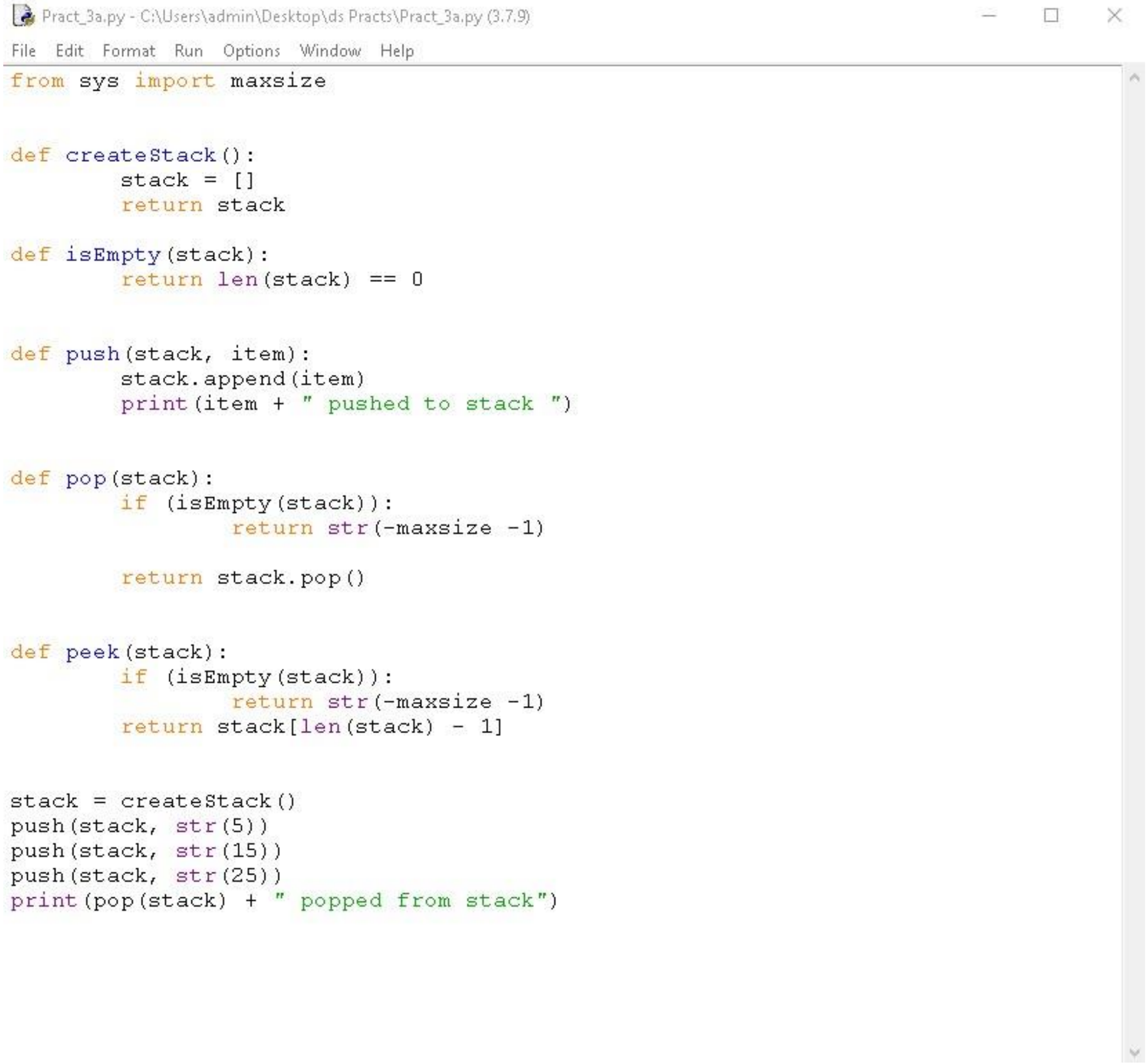
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_2.py =============
-1
6
7
search the specific value :
found Value :  -2
Display the values one by one :
-2
-1
5
-2
4
3
2
1
Himanshu
abc
-1
6
7
peek (End  Value) : 7
treverse the values :
['-2', '-1', '5', '-2', '4', '3', '2', '1', 'Himanshu', 'abc', '-1', '6', '7']
Shotting the values :
['-1', '-1', '-2', '-2', '1', '2', '3', '4', '5', '6', '7', 'Himanshu', 'abc']
Reversing  the values :
['abc', 'Himanshu', '7', '6', '5', '4', '3', '2', '1', '-2', '-2', '-1', '-1']
remove value from particular index which is defined earlier
['-1', '-2', '-2', '1', '2', '3', '4', '5', '6', '7', 'Himanshu', 'abc']
merge
['1', '1', '2', '2', '3', '3', '4', '4', '6', 'Himanshu', 'abc']
>>> 
```

PRACTICAL NO: -3A

AIM: -3a) Perform Stack Operations using Array Implementation

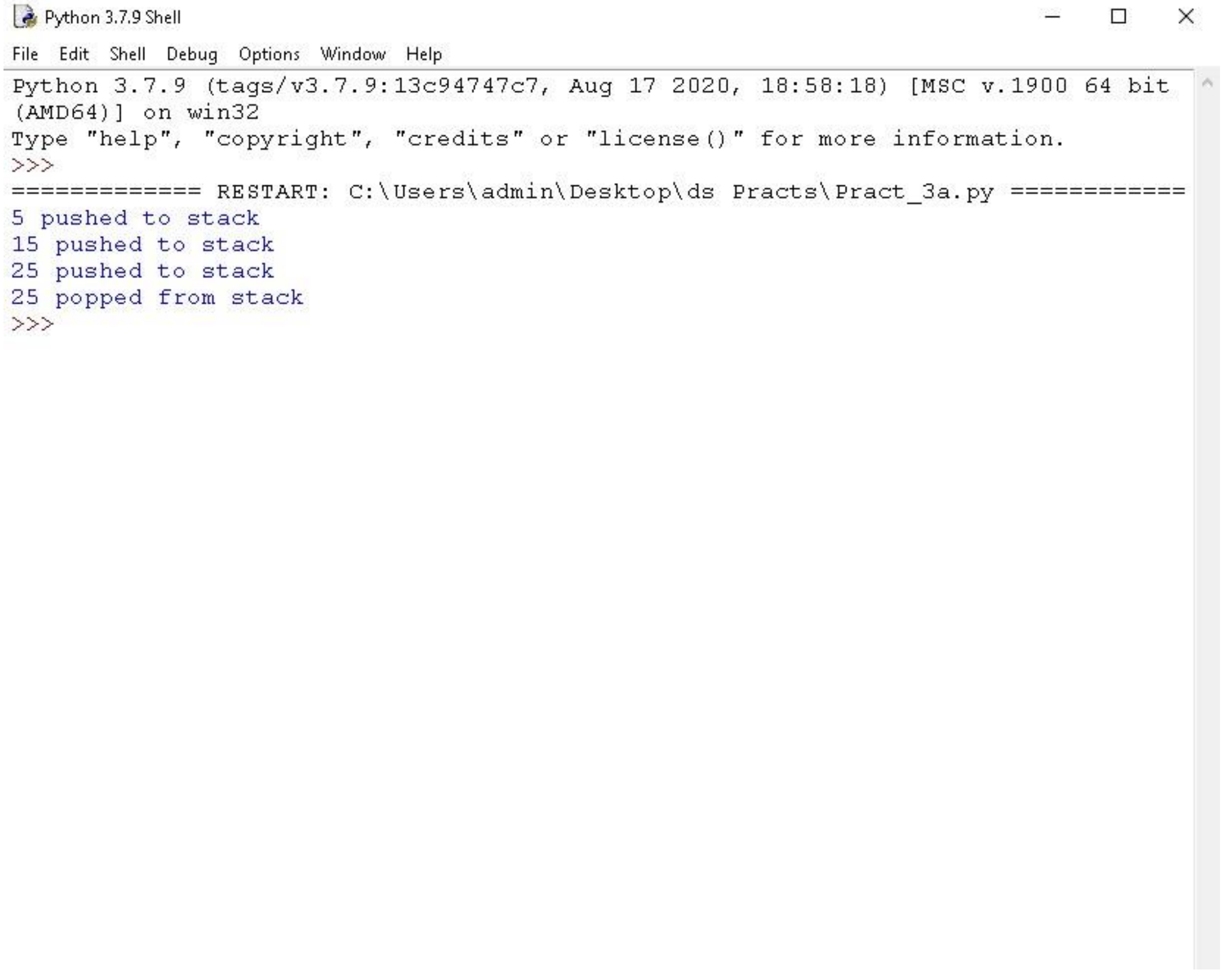
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_3a.py

CODE:


Pract_3a.py - C:\Users\admin\Desktop\ds Practs\Pract_3a.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
from sys import maxsize


def createStack():
        stack = []
        return stack

def isEmpty(stack):
        return len(stack) == 0


def push(stack, item):
        stack.append(item)
        print(item + " pushed to stack ")


def pop(stack):
        if (isEmpty(stack)):
                return str(-maxsize -1)

        return stack.pop()


def peek(stack):
        if (isEmpty(stack)):
                return str(-maxsize -1)
        return stack[len(stack) - 1]


stack = createStack()
push(stack, str(5))
push(stack, str(15))
push(stack, str(25))
print(pop(stack) + " popped from stack")
```

OUTPUT:



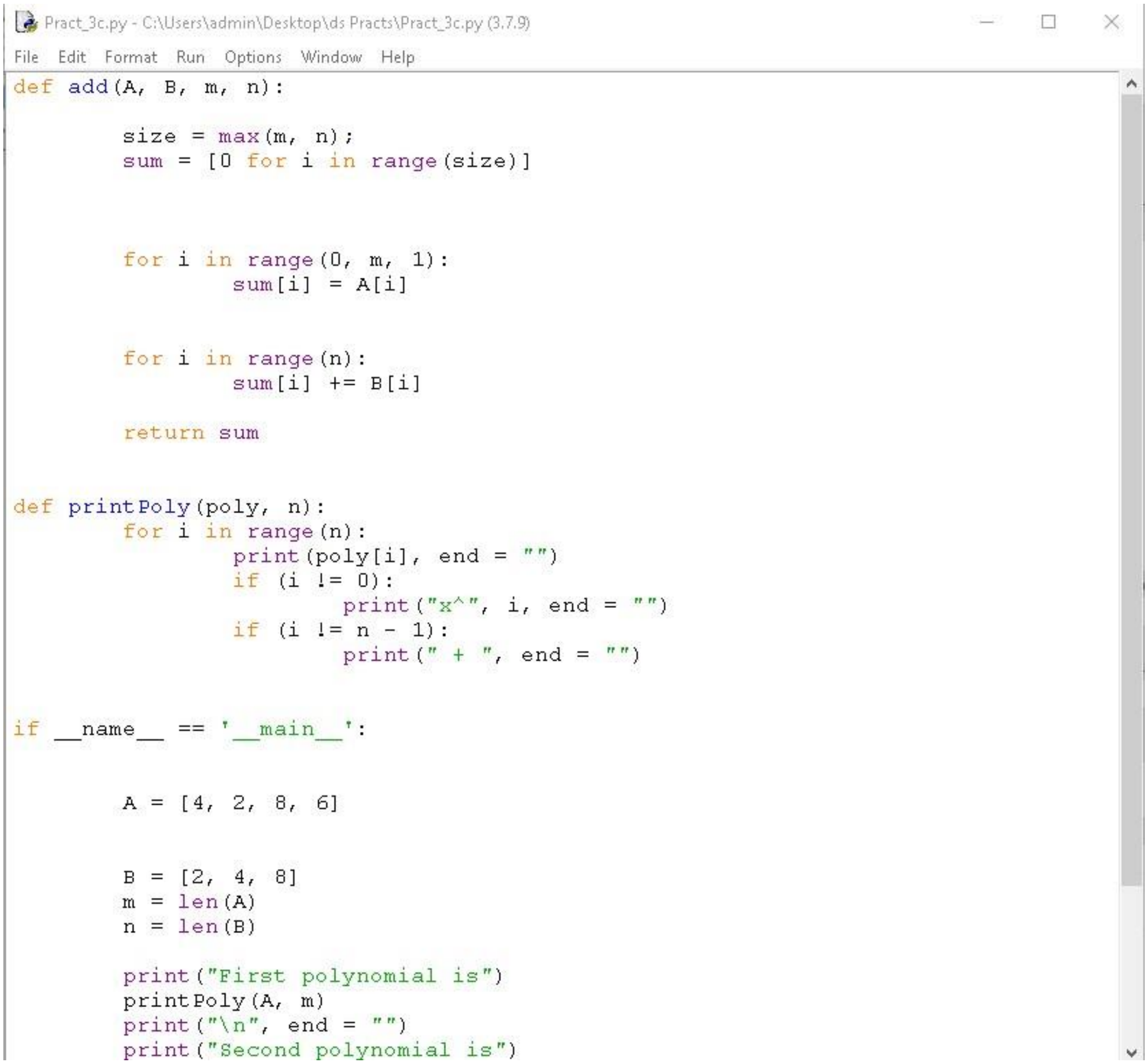
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_3a.py ============
5 pushed to stack
15 pushed to stack
25 pushed to stack
25 popped from stack
>>>
```

PRACTICAL NO: -3C

AIM: -3c) WAP to scan a polynomial using linked list and add two polynomials.

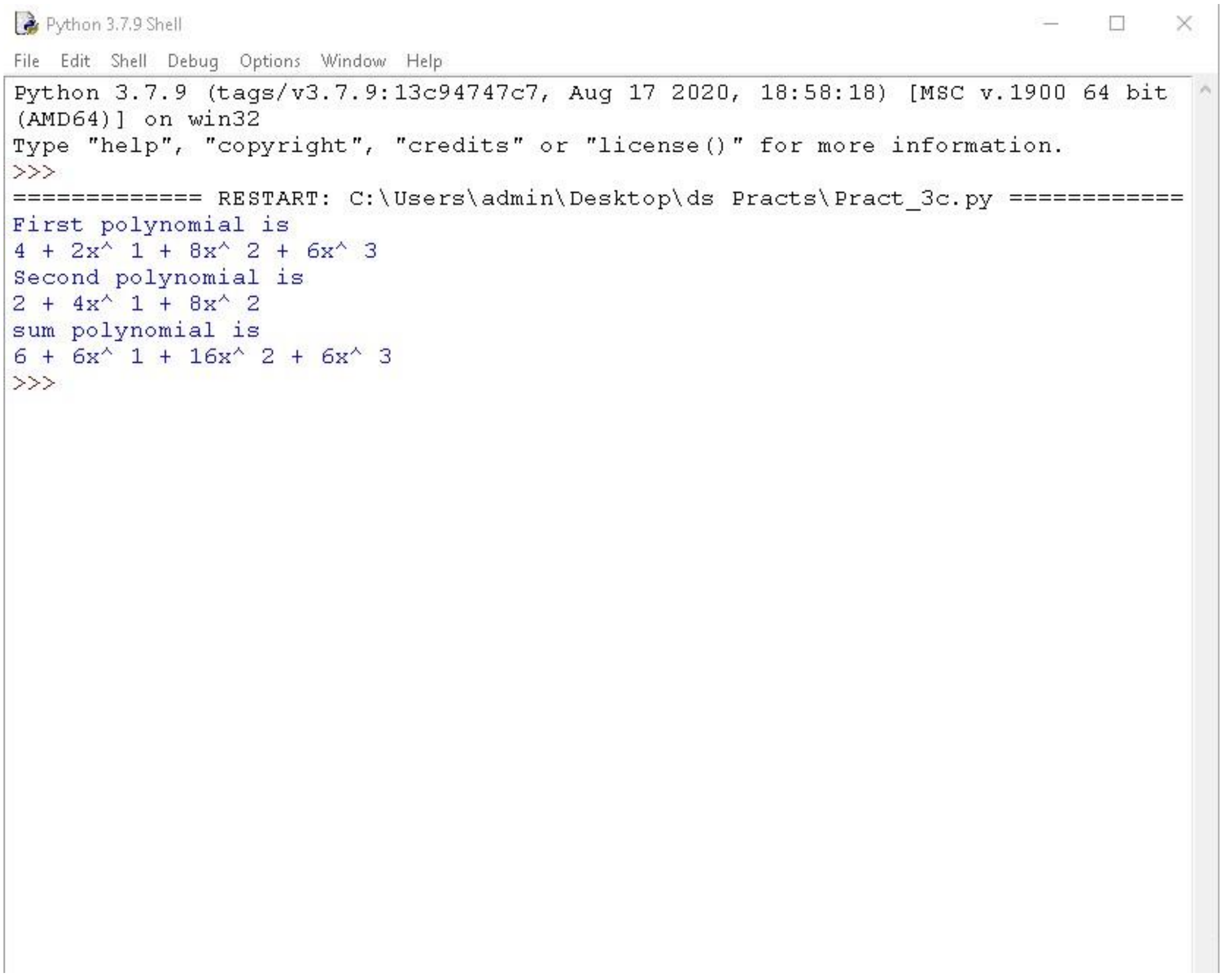
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_3c.py

CODE:


Pract_3c.py - C:\Users\admin\Desktop\ds Practs\Pract_3c.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
def add(A, B, m, n):

        size = max(m, n);
        sum = [0 for i in range(size)]



        for i in range(0, m, 1):
                sum[i] = A[i]


        for i in range(n):
                sum[i] += B[i]

        return sum


def printPoly(poly, n):
        for i in range(n):
                print(poly[i], end = "")
                if (i != 0):
                        print("x^", i, end = "")
                if (i != n - 1):
                        print(" + ", end = "")


if __name__ == '__main__':


        A = [4, 2, 8, 6]


        B = [2, 4, 8]
        m = len(A)
        n = len(B)

        print("First polynomial is")
        printPoly(A, m)
        print("\n", end = "")
        print("Second polynomial is")
```

OUTPUT:



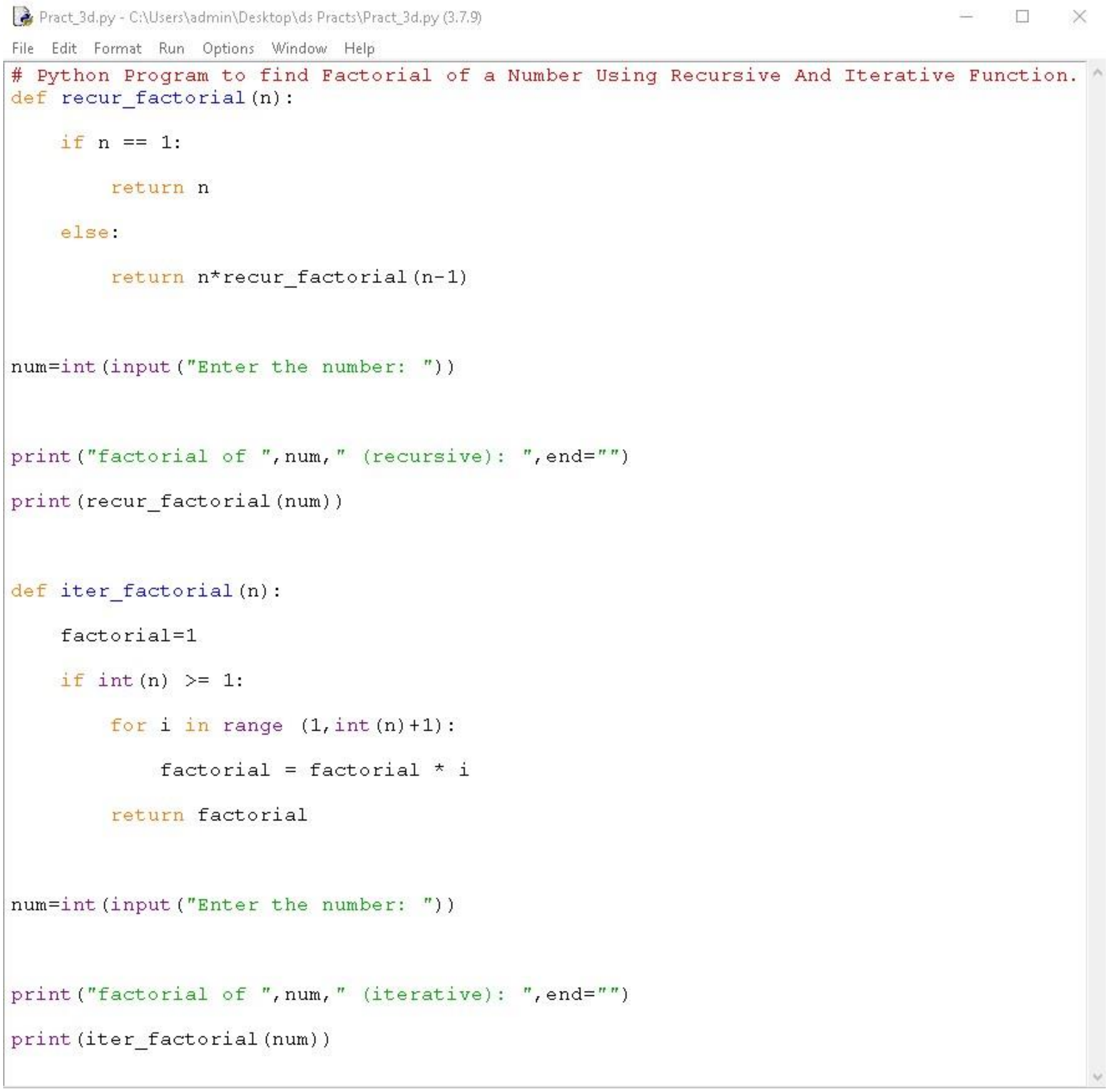
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_3c.py ============
First polynomial is
4 + 2x^ 1 + 8x^ 2 + 6x^ 3
Second polynomial is
2 + 4x^ 1 + 8x^ 2
sum polynomial is
6 + 6x^ 1 + 16x^ 2 + 6x^ 3
>>>
```

PRACTICAL NO: -3D

AIM: -3d) WAP to Calculate Factorial and to Compute the Factors of the given no i) Using Recursion. ii) Using Iteration.

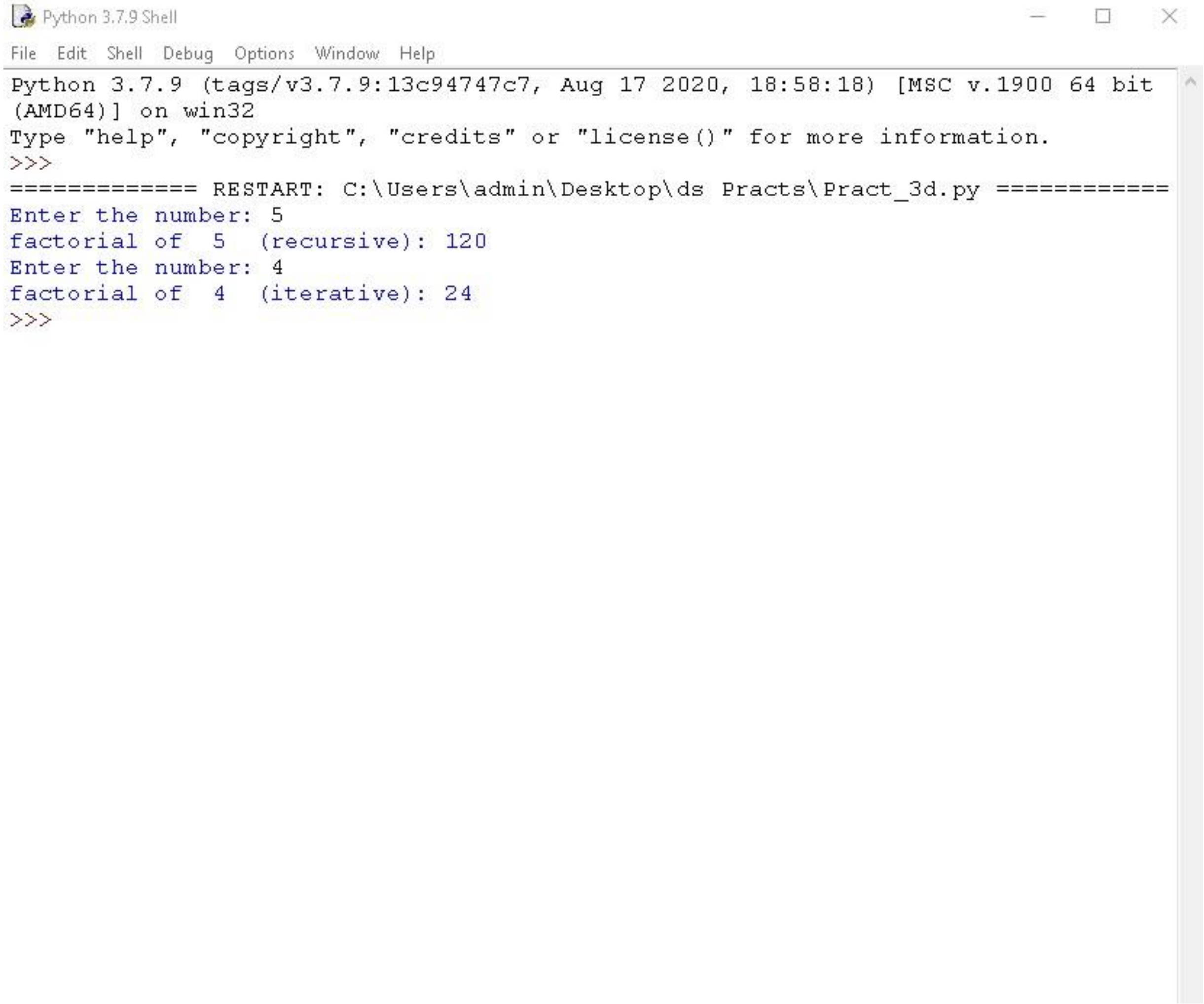
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_3d.py

CODE:


Pract_3d.py - C:\Users\admin\Desktop\ds Practs\Pract_3d.py (3.7.9)
File Edit Format Run Options Window Help

```
# Python Program to find Factorial of a Number Using Recursive And Iterative Function.
def recur_factorial(n):

    if n == 1:

        return n

    else:

        return n*recur_factorial(n-1)


num=int(input("Enter the number:  "))


print("factorial of ",num," (recursive):  ",end="")

print(recur_factorial(num))


def iter_factorial(n):

    factorial=1

    if int(n) >= 1:

        for i in range (1,int(n)+1):

            factorial = factorial * i

        return factorial


num=int(input("Enter the number:  "))


print("factorial of ",num," (iterative):  ",end="")

print(iter_factorial(num))
```

OUTPUT:



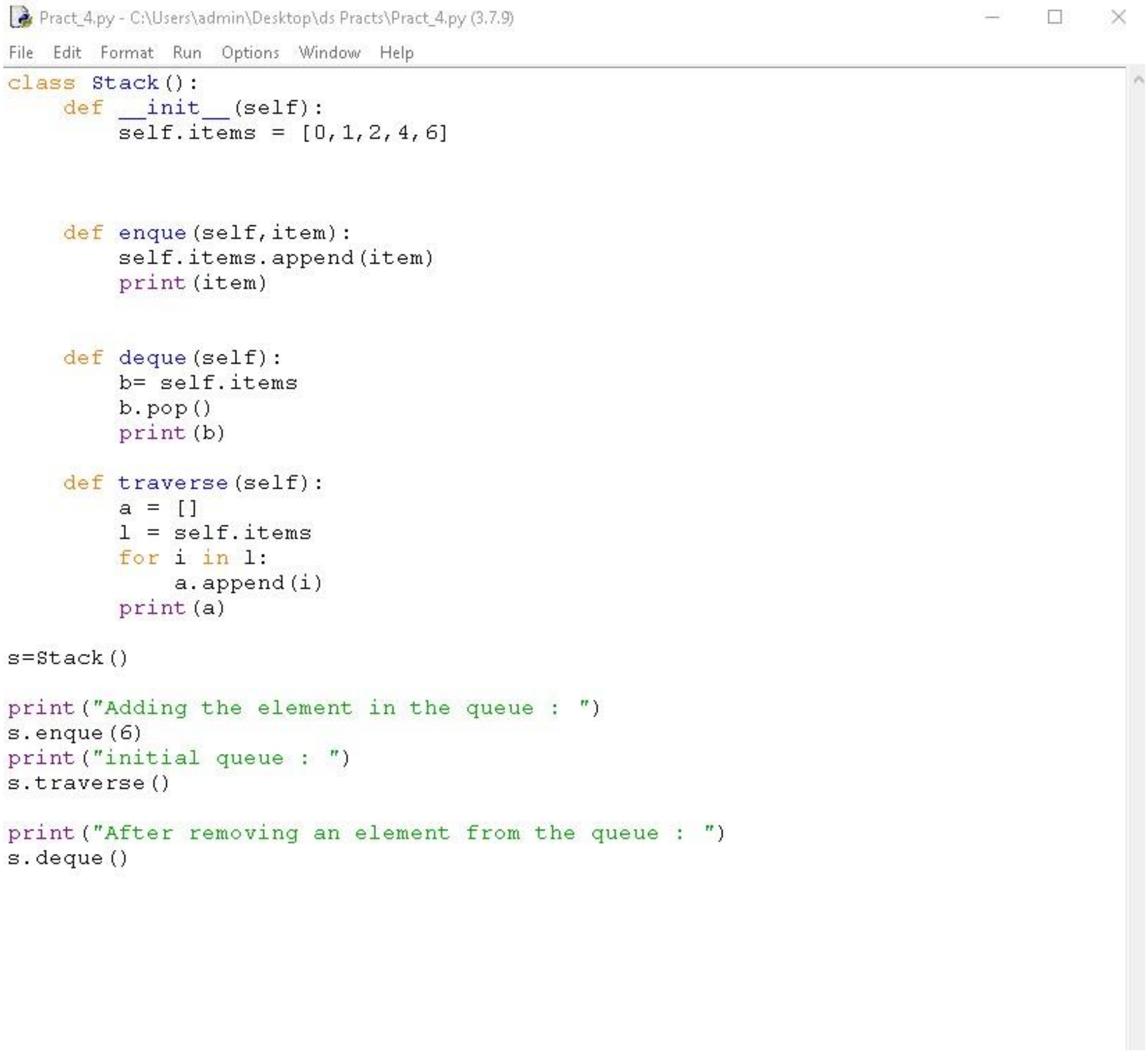
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_3d.py ============
Enter the number: 5
factorial of  5  (recursive): 120
Enter the number: 4
factorial of  4  (iterative): 24
>>>
```

PRACTICAL NO: -4

Aim: 4) Perform Queues Operations using Circular Array Implementation.

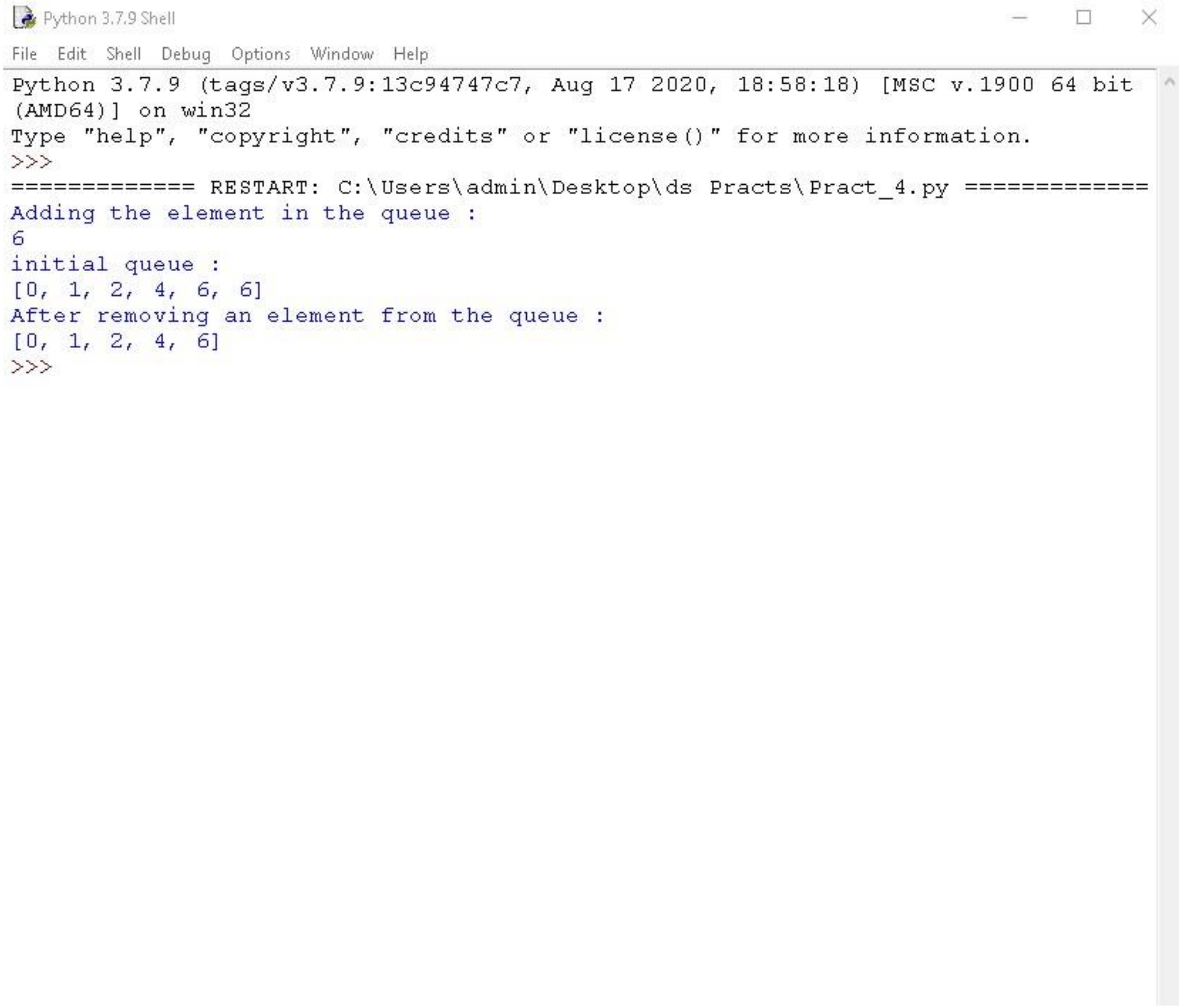
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_4.py

CODE:


Pract_4.py - C:\Users\admin\Desktop\ds Practs\Pract_4.py (3.7.9)
File Edit Format Run Options Window Help

```
class Stack():
    def __init__(self):
        self.items = [0,1,2,4,6]



    def enque(self,item):
        self.items.append(item)
        print(item)


    def deque(self):
        b= self.items
        b.pop()
        print(b)

    def traverse(self):
        a = []
        l = self.items
        for i in l:
            a.append(i)
        print(a)

s=Stack()

print("Adding the element in the queue :  ")
s.enque(6)
print("initial queue :  ")
s.traverse()

print("After removing an element from the queue :  ")
s.deque()
```

OUTPUT:



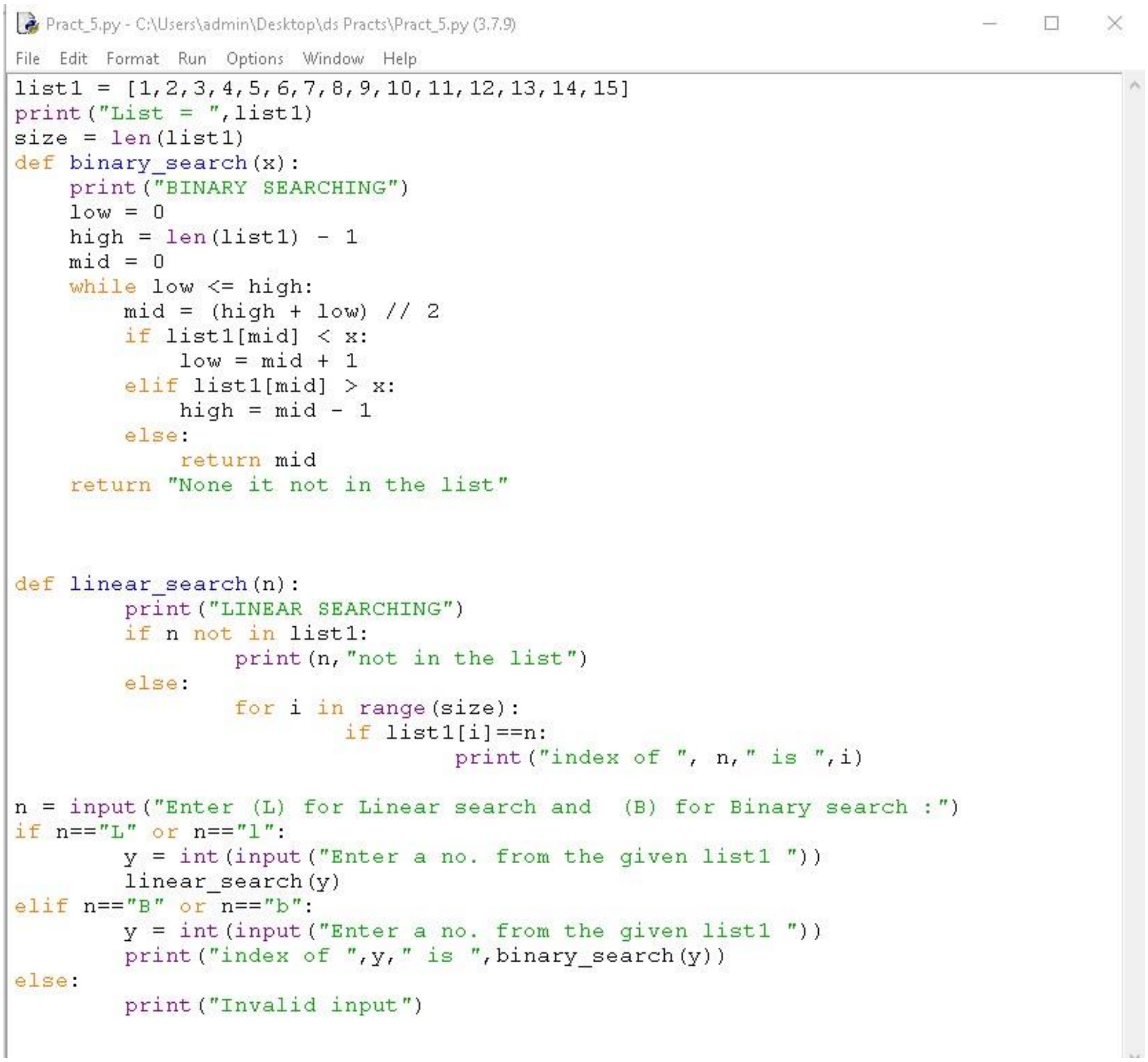
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_4.py =============
Adding the element in the queue :
6
initial queue :
[0, 1, 2, 4, 6, 6]
After removing an element from the queue :
[0, 1, 2, 4, 6]
>>>
```

PRACTICAL NO: -5

AIM: -5) Write a program to search an element from a given list. Give the user option to perform Liner or Binary Search.

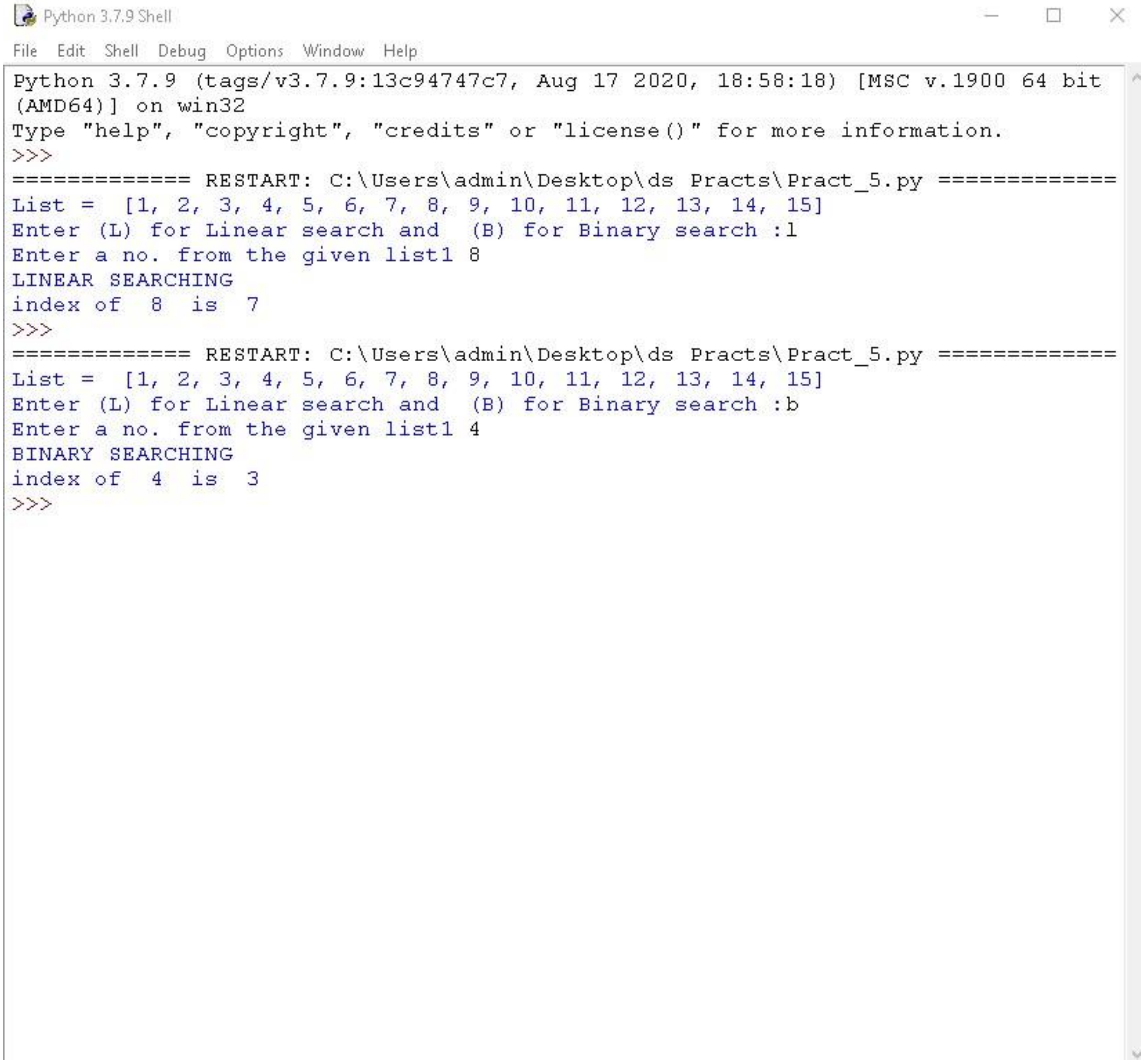
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_5.py

CODE:


Pract_5.py - C:\Users\admin\Desktop\ds Practs\Pract_5.py (3.7.9)
File Edit Format Run Options Window Help

```python
list1 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
print("List = ",list1)
size = len(list1)
def binary_search(x):
    print("BINARY SEARCHING")
    low = 0
    high = len(list1) - 1
    mid = 0
    while low <= high:
        mid = (high + low) // 2
        if list1[mid] < x:
            low = mid + 1
        elif list1[mid] > x:
            high = mid - 1
        else:
            return mid
    return "None it not in the list"



def linear_search(n):
        print("LINEAR SEARCHING")
        if n not in list1:
                print(n,"not in the list")
        else:
                for i in range(size):
                        if list1[i]==n:
                                print("index of ", n," is ",i)

n = input("Enter (L) for Linear search and  (B) for Binary search :")
if n=="L" or n=="l":
        y = int(input("Enter a no. from the given list1 "))
        linear_search(y)
elif n=="B" or n=="b":
        y = int(input("Enter a no. from the given list1 "))
        print("index of ",y," is ",binary_search(y))
else:
        print("Invalid input")
```

OUTPUT:



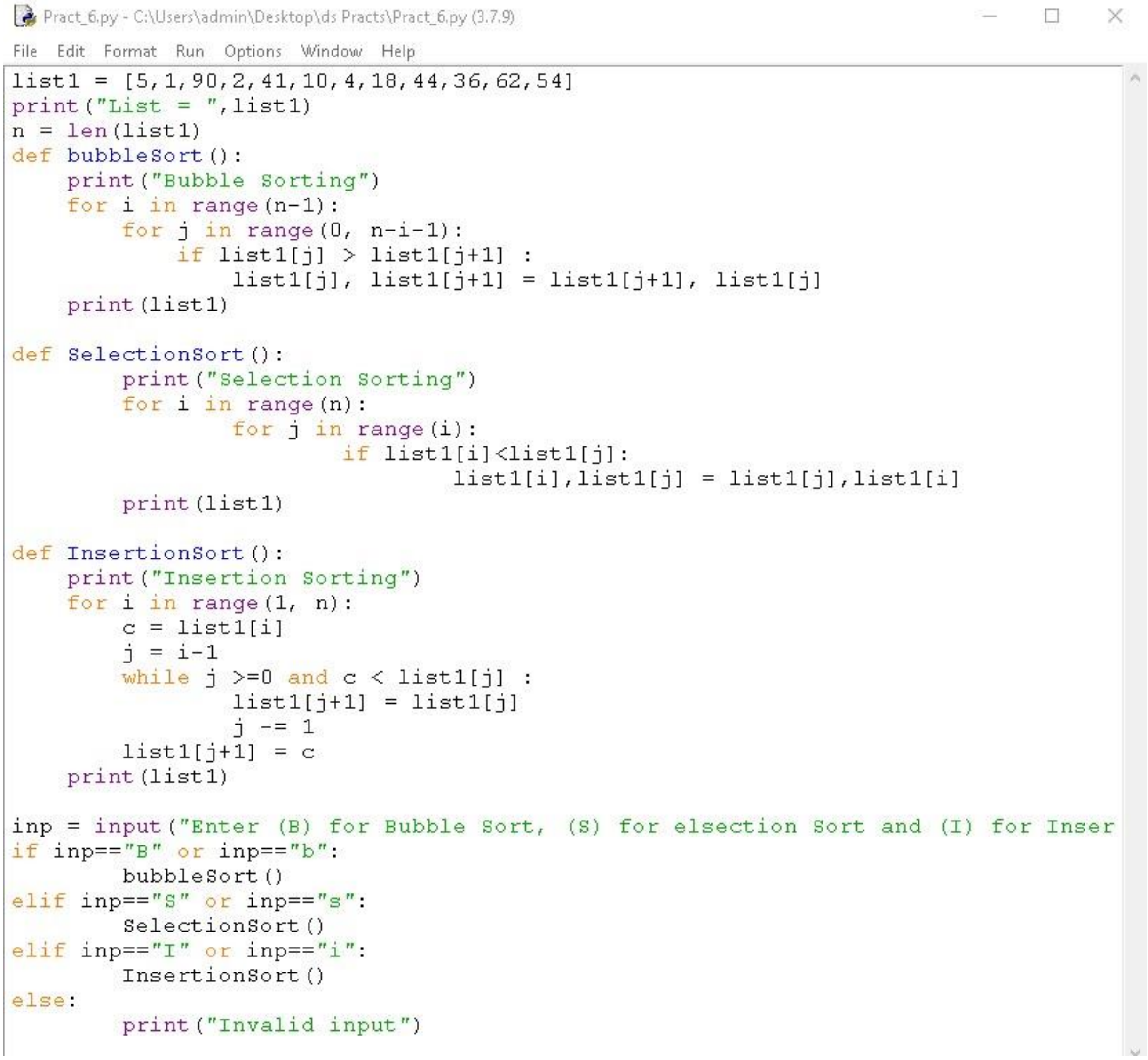
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_5.py =============
List =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
Enter (L) for Linear search and  (B) for Binary search :l
Enter a no. from the given list1 8
LINEAR SEARCHING
index of  8  is  7
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_5.py =============
List =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
Enter (L) for Linear search and  (B) for Binary search :b
Enter a no. from the given list1 4
BINARY SEARCHING
index of  4  is  3
>>>
```

PRACTICAL NO: -6

AIM: -6) WAP to sort a list of Elements. Give the user option to perform sorting using Insertion sort, Bubble sort or Selection sort.

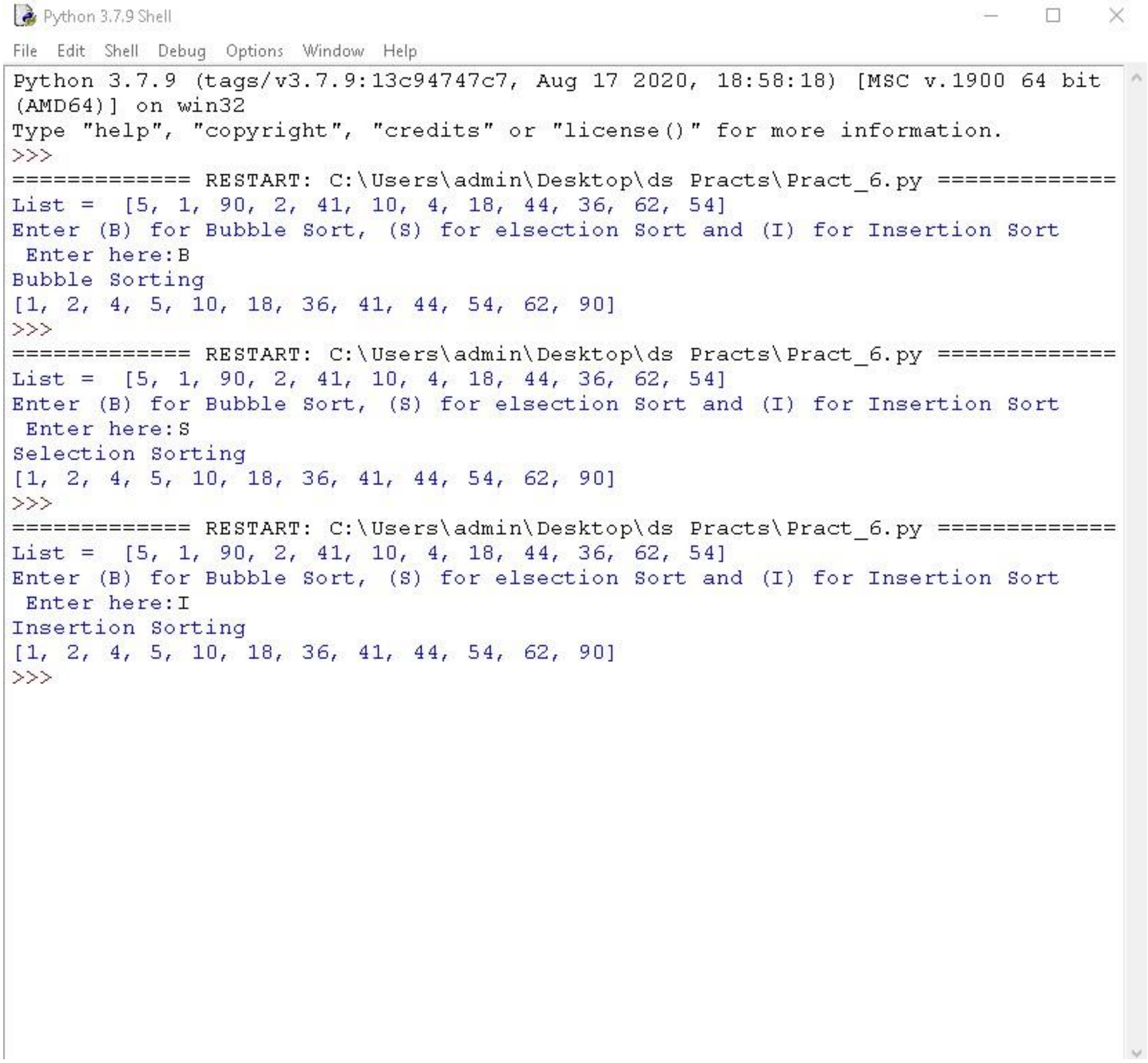
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_6.py

CODE:


Pract_6.py - C:\Users\admin\Desktop\ds Practs\Pract_6.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
list1 = [5,1,90,2,41,10,4,18,44,36,62,54]
print("List = ",list1)
n = len(list1)
def bubbleSort():
    print("Bubble Sorting")
    for i in range(n-1):
        for j in range(0, n-i-1):
            if list1[j] > list1[j+1] :
                list1[j], list1[j+1] = list1[j+1], list1[j]
    print(list1)

def SelectionSort():
        print("Selection Sorting")
        for i in range(n):
                for j in range(i):
                        if list1[i]<list1[j]:
                                list1[i],list1[j] = list1[j],list1[i]
        print(list1)

def InsertionSort():
    print("Insertion Sorting")
    for i in range(1, n):
        c = list1[i]
        j = i-1
        while j >=0 and c < list1[j] :
                list1[j+1] = list1[j]
                j -= 1
        list1[j+1] = c
    print(list1)

inp = input("Enter (B) for Bubble Sort, (S) for elsection Sort and (I) for Inser
if inp=="B" or inp=="b":
        bubbleSort()
elif inp=="S" or inp=="s":
        SelectionSort()
elif inp=="I" or inp=="i":
        InsertionSort()
else:
        print("Invalid input")
```

OUTPUT:



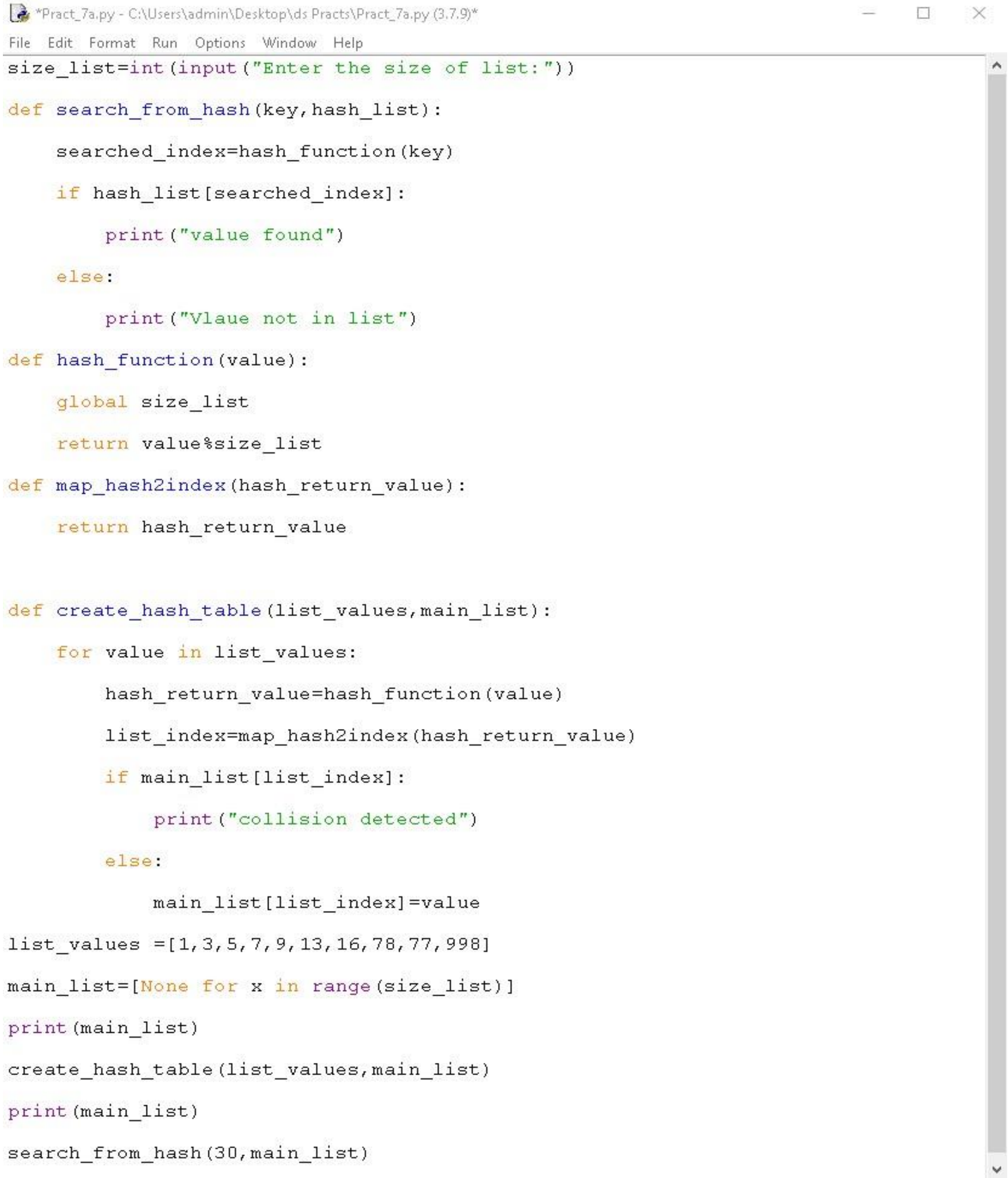
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_6.py =============
List =  [5, 1, 90, 2, 41, 10, 4, 18, 44, 36, 62, 54]
Enter (B) for Bubble Sort, (S) for elsection Sort and (I) for Insertion Sort
 Enter here:B
Bubble Sorting
[1, 2, 4, 5, 10, 18, 36, 41, 44, 54, 62, 90]
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_6.py =============
List =  [5, 1, 90, 2, 41, 10, 4, 18, 44, 36, 62, 54]
Enter (B) for Bubble Sort, (S) for elsection Sort and (I) for Insertion Sort
 Enter here:S
Selection Sorting
[1, 2, 4, 5, 10, 18, 36, 41, 44, 54, 62, 90]
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_6.py =============
List =  [5, 1, 90, 2, 41, 10, 4, 18, 44, 36, 62, 54]
Enter (B) for Bubble Sort, (S) for elsection Sort and (I) for Insertion Sort
 Enter here:I
Insertion Sorting
[1, 2, 4, 5, 10, 18, 36, 41, 44, 54, 62, 90]
>>>
```

PRACTICAL NO: -7A

AIM: -7a) Write a Program to Implement the Collision Technique.

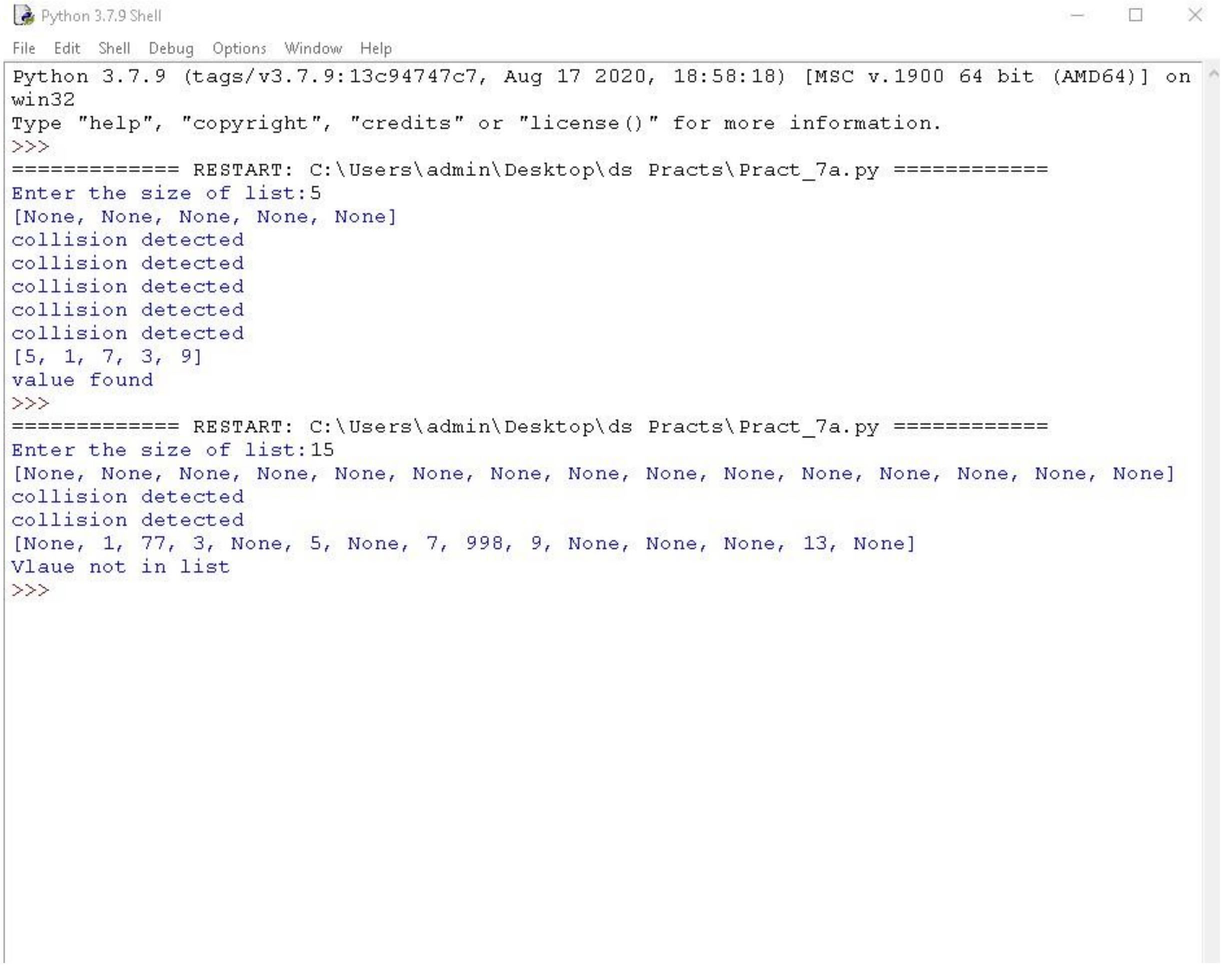
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_7a.py

CODE:


Pract_7a.py - C:\Users\admin\Desktop\ds Practs\Pract_7a.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
size_list=int(input("Enter the size of list:"))
def search_from_hash(key,hash_list):
    searched_index=hash_function(key)
    if hash_list[searched_index]:
        print("value found")
    else:
        print("Vlaue not in list")
def hash_function(value):
    global size_list
    return value%size_list
def map_hash2index(hash_return_value):
    return hash_return_value


def create_hash_table(list_values,main_list):
    for value in list_values:
        hash_return_value=hash_function(value)
        list_index=map_hash2index(hash_return_value)
        if main_list[list_index]:
            print("collision detected")
        else:
            main_list[list_index]=value
list_values =[1,3,5,7,9,13,16,78,77,998]
main_list=[None for x in range(size_list)]
print(main_list)
create_hash_table(list_values,main_list)
print(main_list)
search_from_hash(30,main_list)
```

OUTPUT:



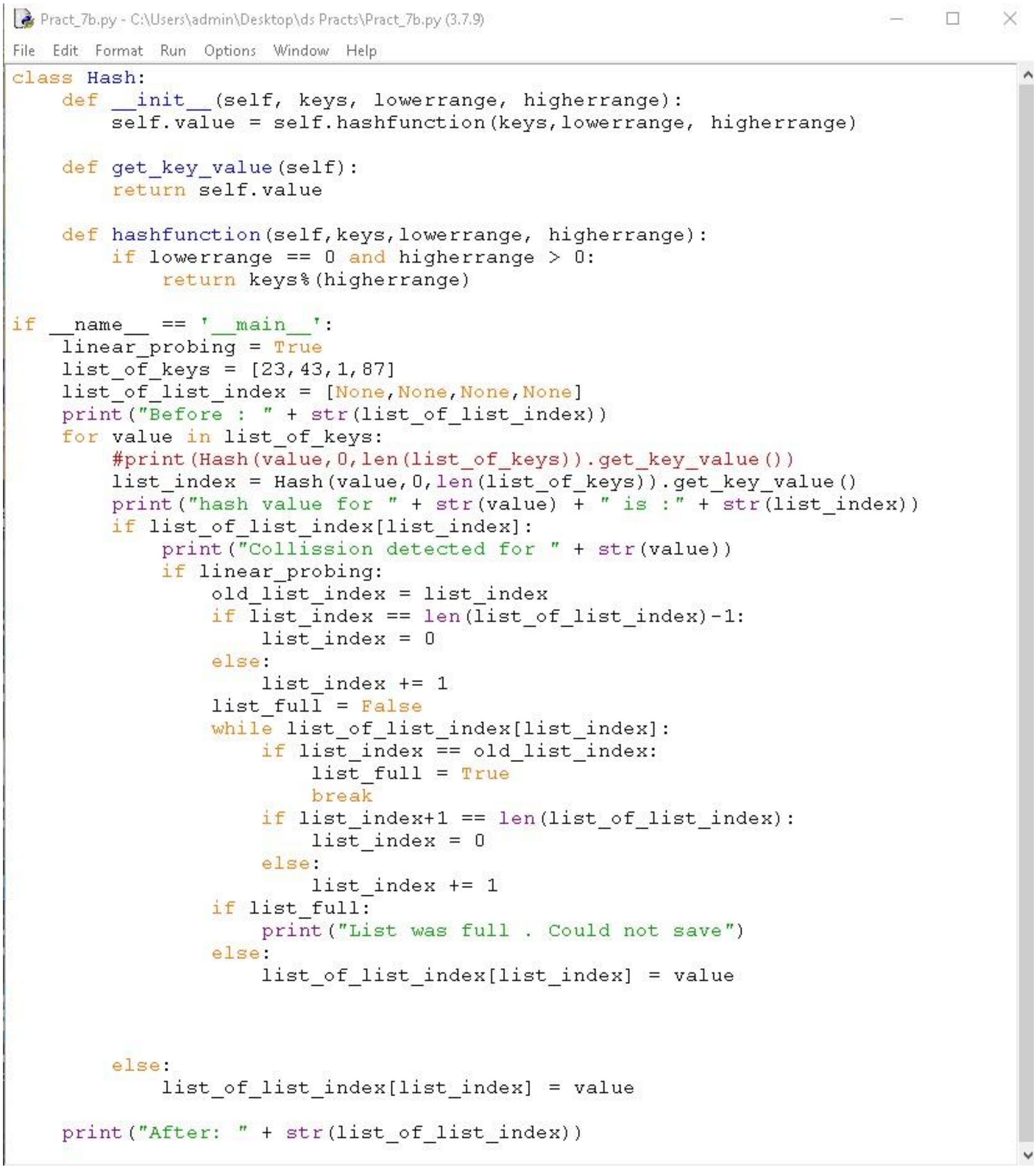
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_7a.py ============
Enter the size of list:5
[None, None, None, None, None]
collision detected
collision detected
collision detected
collision detected
collision detected
[5, 1, 7, 3, 9]
value found
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_7a.py ============
Enter the size of list:15
[None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]
collision detected
collision detected
[None, 1, 77, 3, None, 5, None, 7, 998, 9, None, None, None, 13, None]
Vlaue not in list
>>>
```

PRACTICAL NO: -7B

AIM: -7b) Write a program to implement the concept of liner Probing.

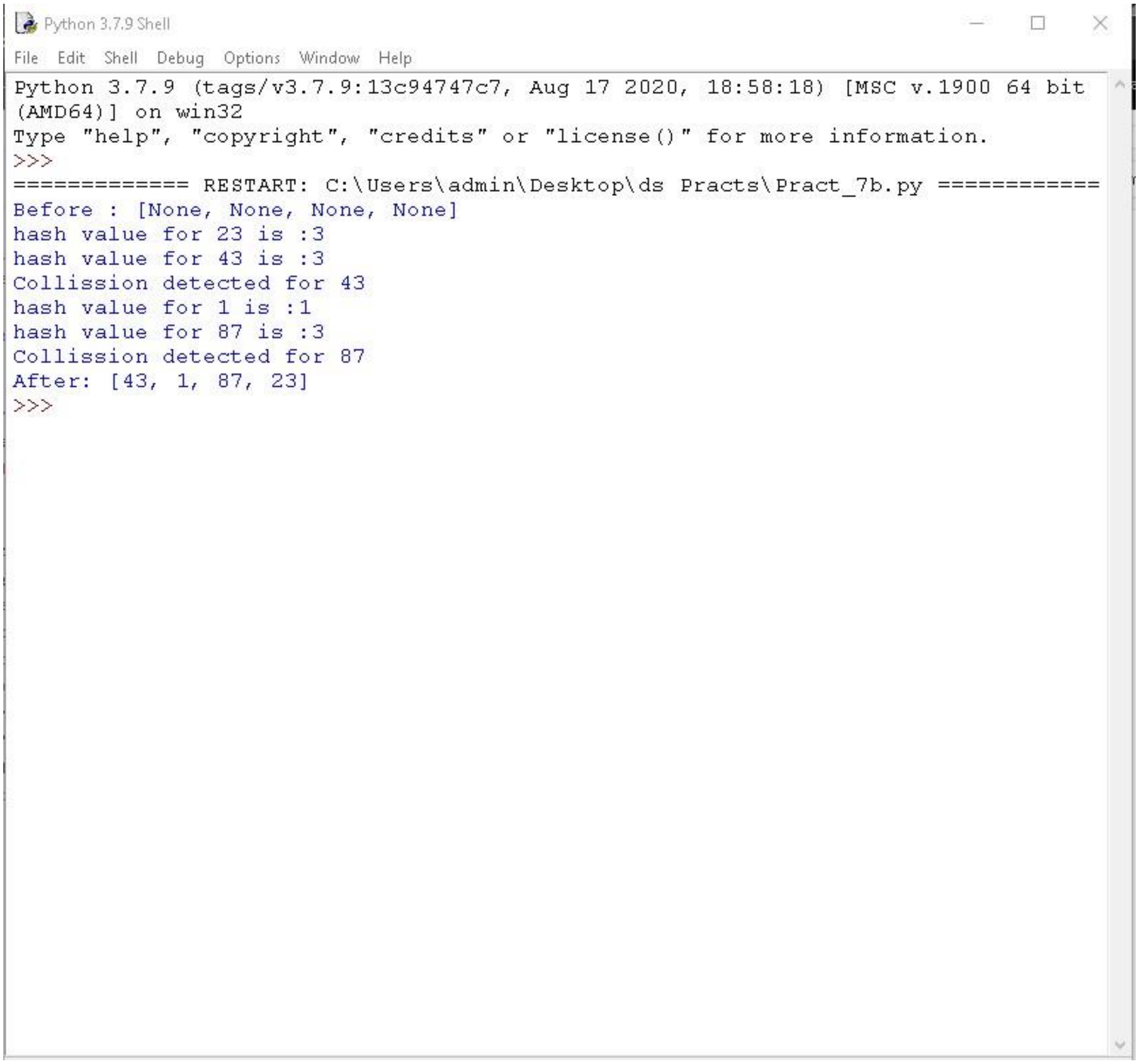
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_7b.py

CODE:


Pract_7b.py - C:\Users\admin\Desktop\ds Practs\Pract_7b.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```
class Hash:
    def __init__(self, keys, lowerrange, higherrange):
        self.value = self.hashfunction(keys,lowerrange, higherrange)

    def get_key_value(self):
        return self.value

    def hashfunction(self,keys,lowerrange, higherrange):
        if lowerrange == 0 and higherrange > 0:
            return keys%(higherrange)

if __name__ == '__main__':
    linear_probing = True
    list_of_keys = [23,43,1,87]
    list_of_list_index = [None,None,None,None]
    print("Before : " + str(list_of_list_index))
    for value in list_of_keys:
        #print(Hash(value,0,len(list_of_keys)).get_key_value())
        list_index = Hash(value,0,len(list_of_keys)).get_key_value()
        print("hash value for " + str(value) + " is :" + str(list_index))
        if list_of_list_index[list_index]:
            print("Collission detected for " + str(value))
            if linear_probing:
                old_list_index = list_index
                if list_index == len(list_of_list_index)-1:
                    list_index = 0
                else:
                    list_index += 1
                list_full = False
                while list_of_list_index[list_index]:
                    if list_index == old_list_index:
                        list_full = True
                        break
                    if list_index+1 == len(list_of_list_index):
                        list_index = 0
                    else:
                        list_index += 1
                if list_full:
                    print("List was full . Could not save")
                else:
                    list_of_list_index[list_index] = value



        else:
            list_of_list_index[list_index] = value

    print("After: " + str(list_of_list_index))
```

OUTPUT:



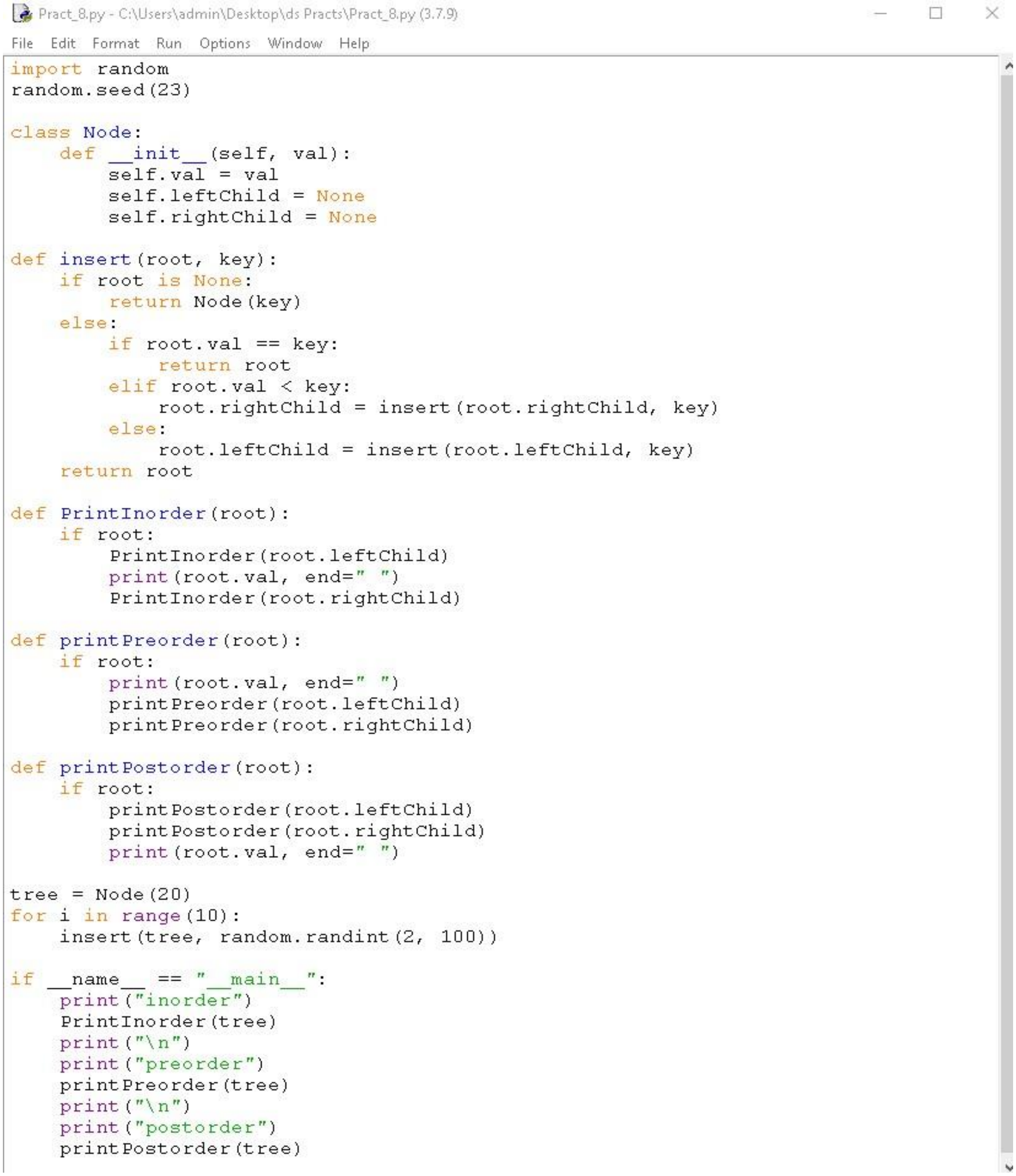
```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_7b.py ============
Before : [None, None, None, None]
hash value for 23 is :3
hash value for 43 is :3
Collission detected for 43
hash value for 1 is :1
hash value for 87 is :3
Collission detected for 87
After: [43, 1, 87, 23]
>>> 
```

PRACTICAL NO: -8

AIM: -8) Write a Program for Inorder, Post order and Pre order traversal of tree.

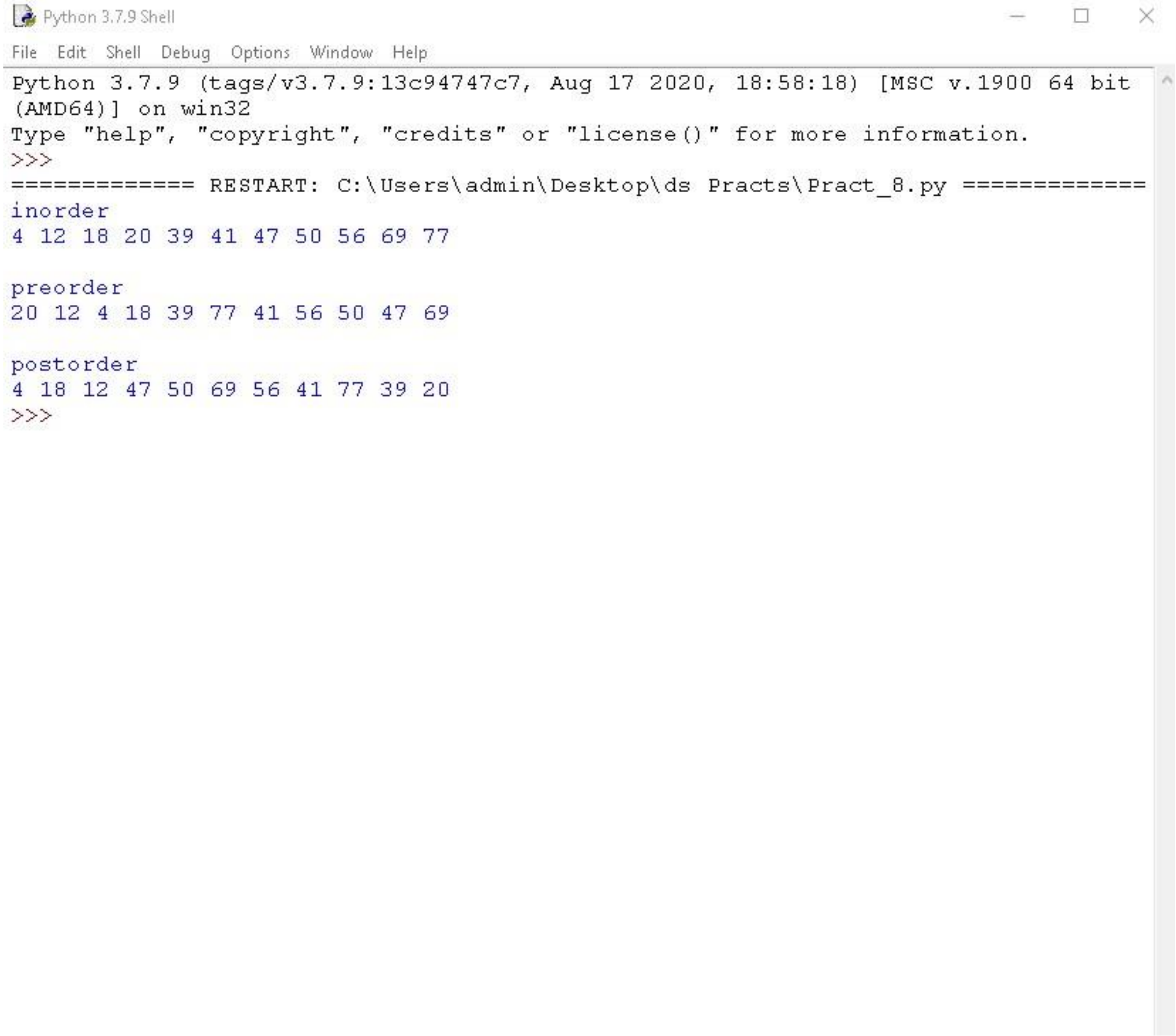
LINK: https://github.com/4068Himanshu/Ds-Practicals/blob/master/Pract_8.py

CODE:


Pract_8.py - C:\Users\admin\Desktop\ds Practs\Pract_8.py (3.7.9)
File
Edit
Format
Run
Options
Window
Help

```python
import random
random.seed(23)

class Node:
    def __init__(self, val):
        self.val = val
        self.leftChild = None
        self.rightChild = None

def insert(root, key):
    if root is None:
        return Node(key)
    else:
        if root.val == key:
            return root
        elif root.val < key:
            root.rightChild = insert(root.rightChild, key)
        else:
            root.leftChild = insert(root.leftChild, key)
    return root

def PrintInorder(root):
    if root:
        PrintInorder(root.leftChild)
        print(root.val, end=" ")
        PrintInorder(root.rightChild)

def printPreorder(root):
    if root:
        print(root.val, end=" ")
        printPreorder(root.leftChild)
        printPreorder(root.rightChild)

def printPostorder(root):
    if root:
        printPostorder(root.leftChild)
        printPostorder(root.rightChild)
        print(root.val, end=" ")

tree = Node(20)
for i in range(10):
    insert(tree, random.randint(2, 100))

if __name__ == "__main__":
    print("inorder")
    PrintInorder(tree)
    print("\n")
    print("preorder")
    printPreorder(tree)
    print("\n")
    print("postorder")
    printPostorder(tree)
```

OUTPUT:



```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: C:\Users\admin\Desktop\ds Practs\Pract_8.py =============
inorder
4 12 18 20 39 41 47 50 56 69 77

preorder
20 12 4 18 39 77 41 56 50 47 69

postorder
4 18 12 47 50 69 56 41 77 39 20
>>>
```