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

CERTIFICATE

Name: Mr.____________________**Hardik D. Vaghasiya**________________________

Roll No:365 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 **Data Structures (Course Code: 2032UISPR)** 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: 365**

Subject: Data Structures

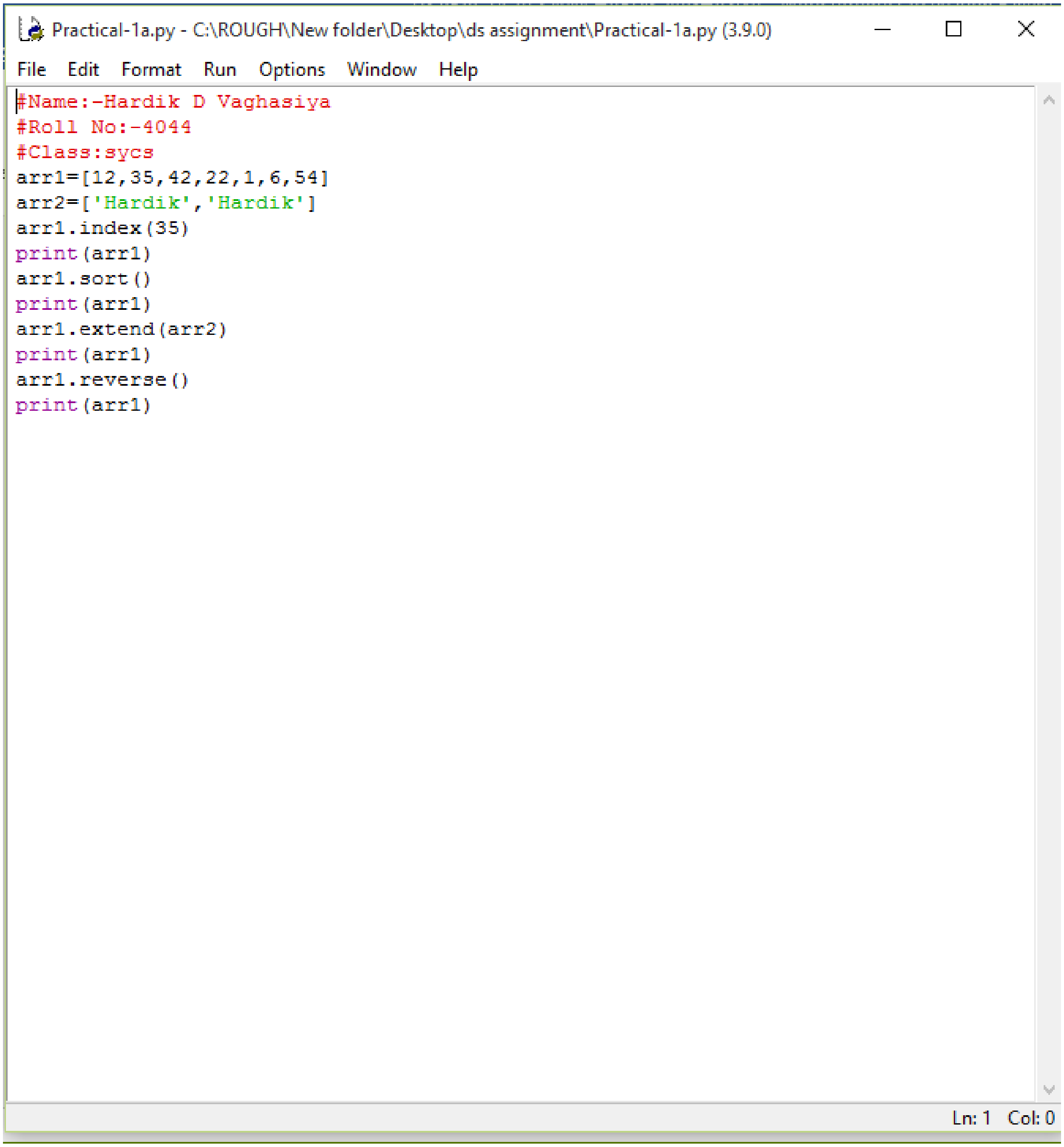
INDEX

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/Hardikvaghasiya/4044-DS/blob/main/Practical-1a

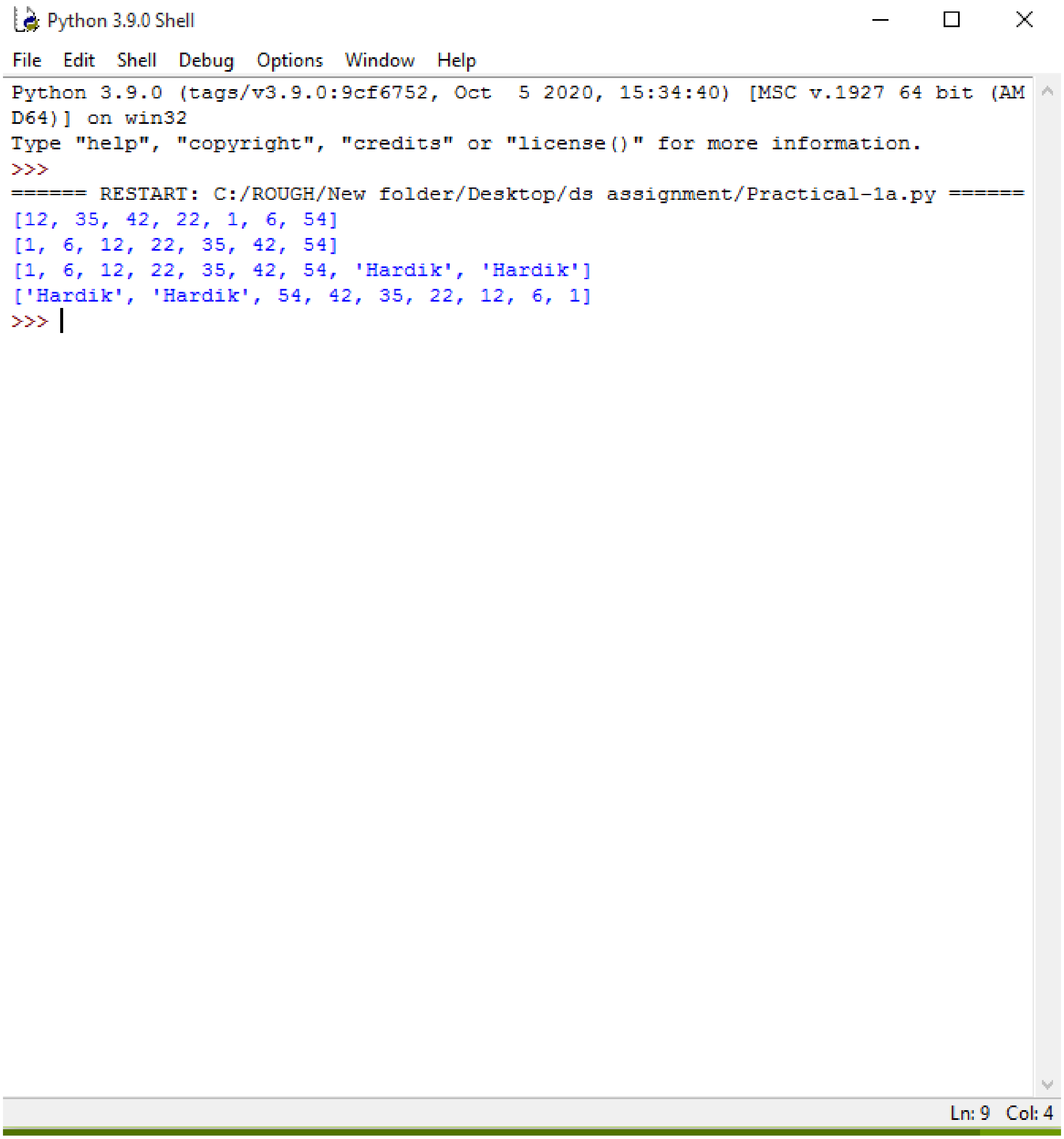
CODE:-



Practical-1a.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-1a.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```
#Name:-Hardik D Vaghasiya
#Roll No:-4044
#Class:sycs
arr1=[12,35,42,22,1,6,54]
arr2=['Hardik','Hardik']
arr1.index(35)
print(arr1)
arr1.sort()
print(arr1)
arr1.extend(arr2)
print(arr1)
arr1.reverse()
print(arr1)
```

OutPut:-



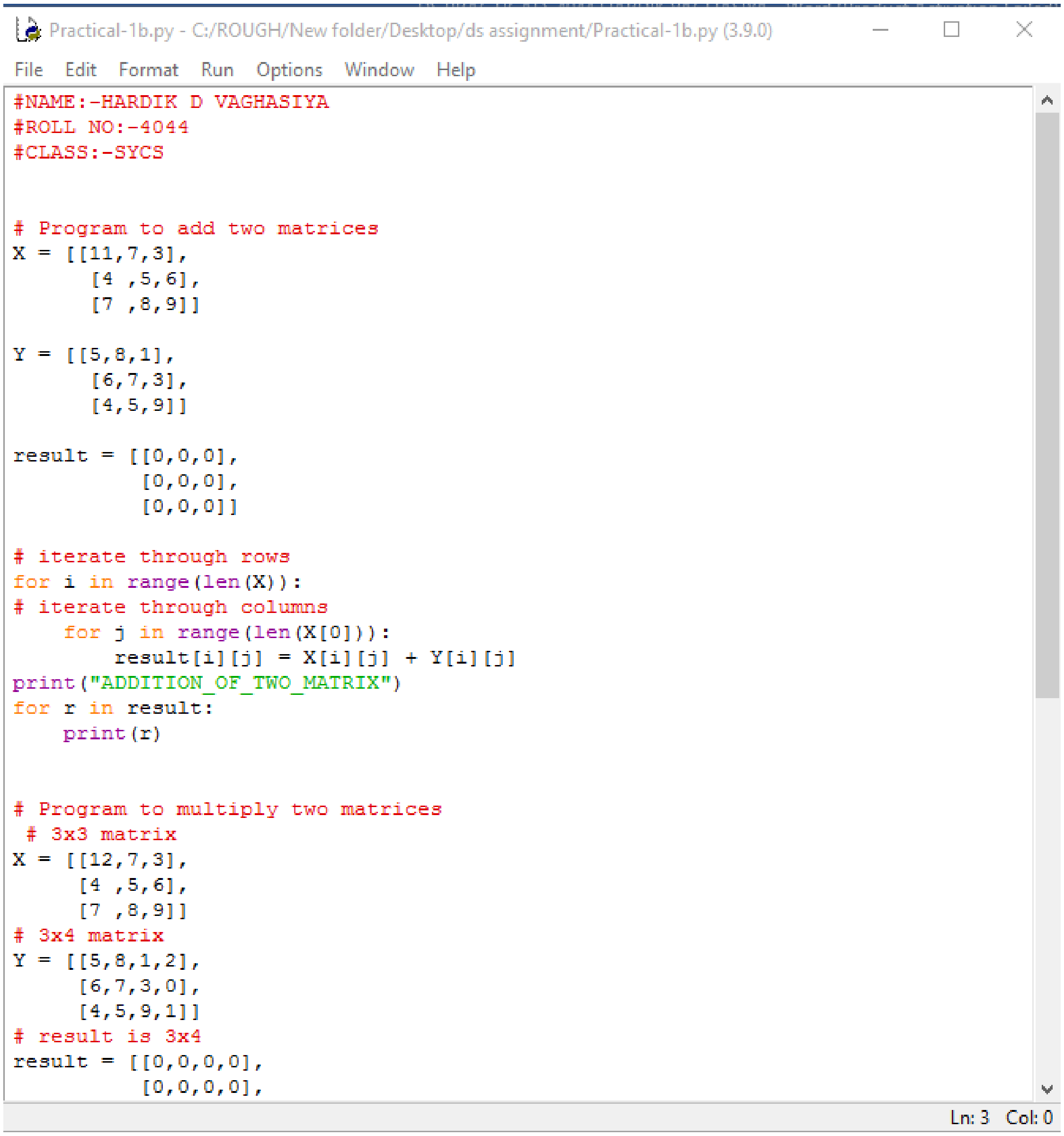
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
====== RESTART: C:/ROUGH/New folder/Desktop/ds assignment/Practical-1a.py ======
[12, 35, 42, 22, 1, 6, 54]
[1, 6, 12, 22, 35, 42, 54]
[1, 6, 12, 22, 35, 42, 54, 'Hardik', 'Hardik']
['Hardik', 'Hardik', 54, 42, 35, 22, 12, 6, 1]
>>> 
```

PRACTICAL NO:-1B

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

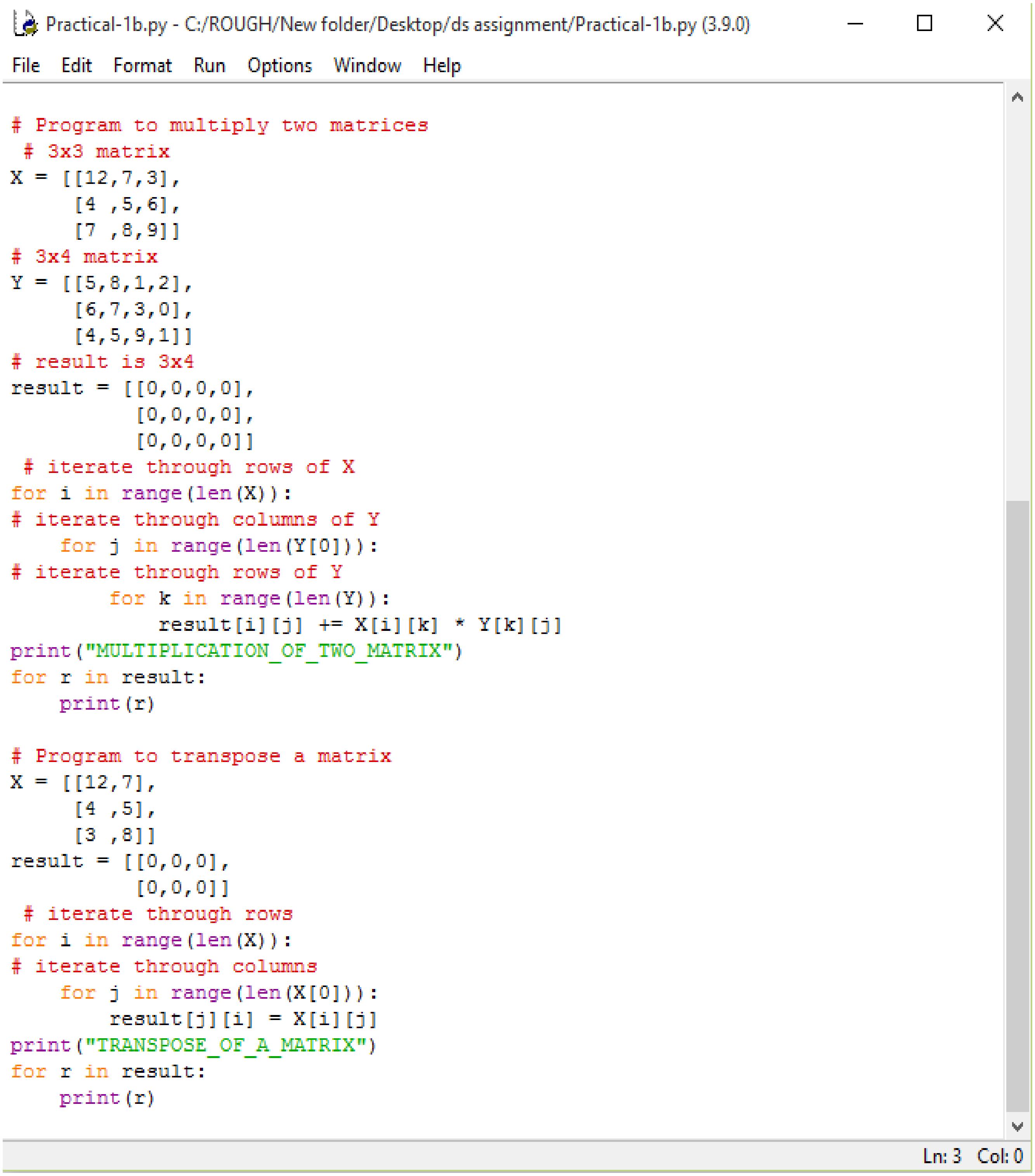
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-1b

CODE:-



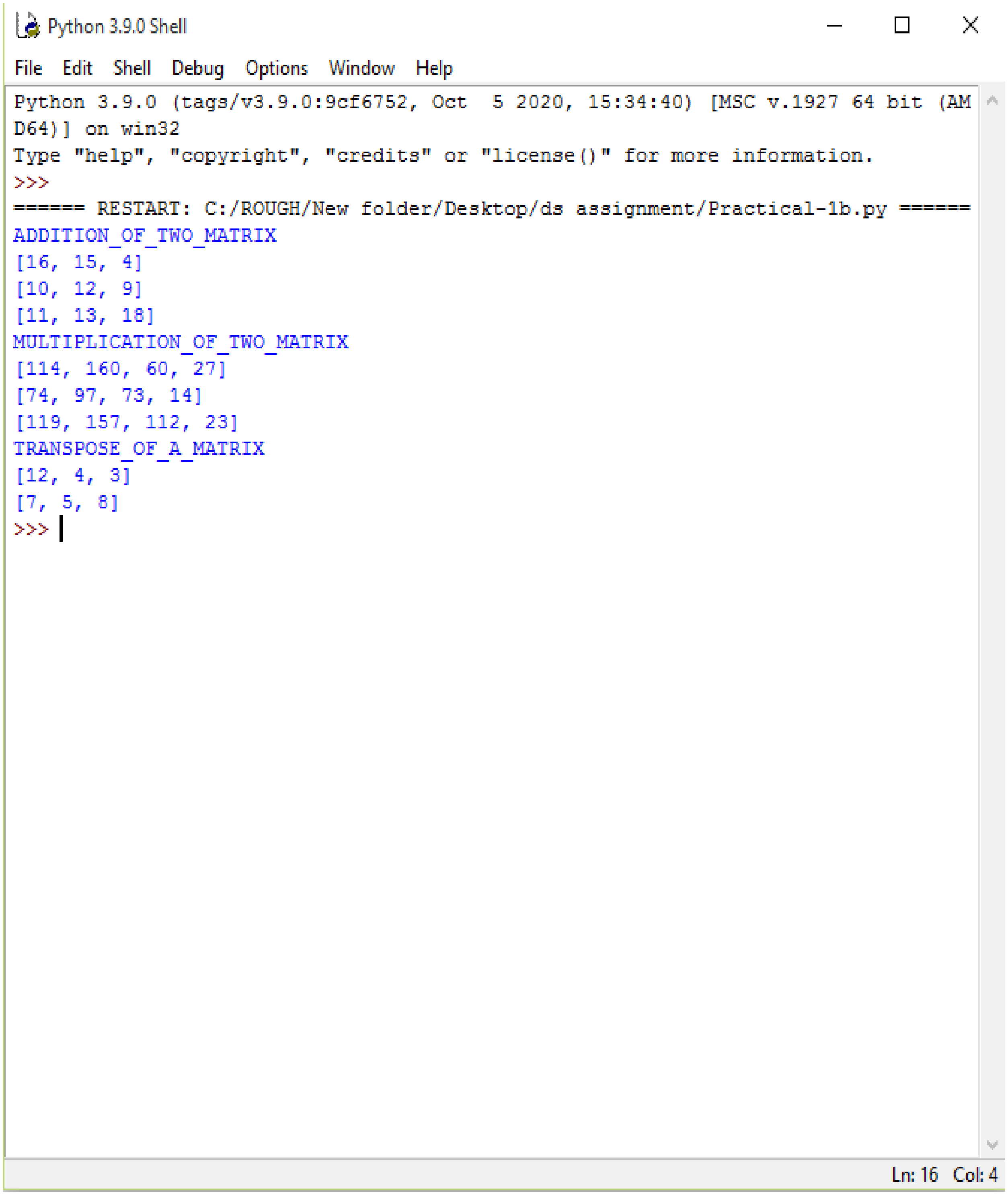
Practical-1b.py - C:/ROUGH/New folder/Desktop/ds assignment/Practical-1b.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 3 Col: 0

```
#NAME:-HARDIK D VAGHASIYA
#ROLL NO:-4044
#CLASS:-SYCS


# 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)):
# iterate through columns
    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],
```


Practical-1b.py - C:/ROUGH/New folder/Desktop/ds assignment/Practical-1b.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 3 Col: 0

```
# 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]]
 # iterate through rows
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:-



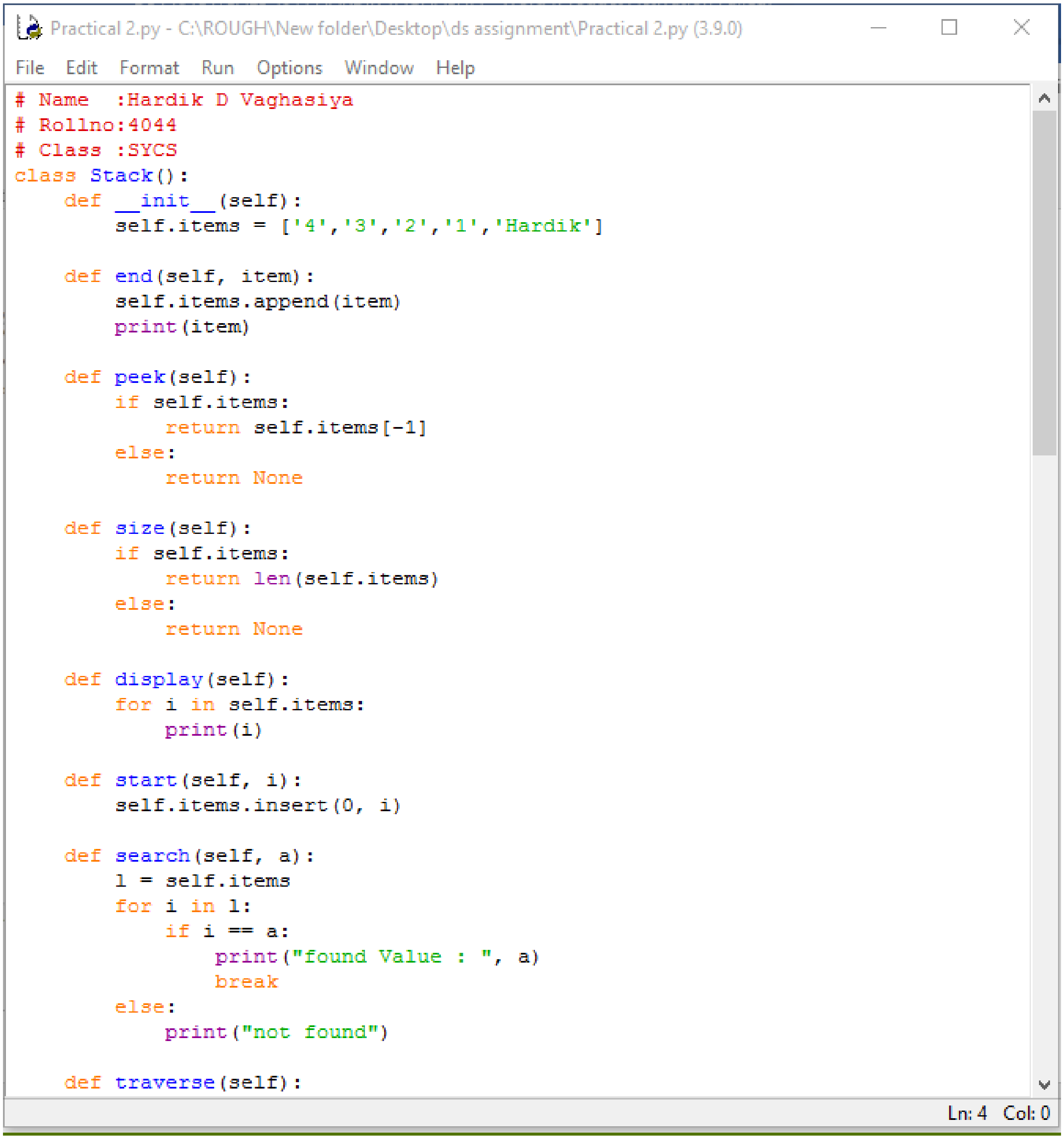
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
====== RESTART: C:/ROUGH/New folder/Desktop/ds assignment/Practical-1b.py ======
ADDITION_OF_TWO_MATRIX
[16, 15, 4]
[10, 12, 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

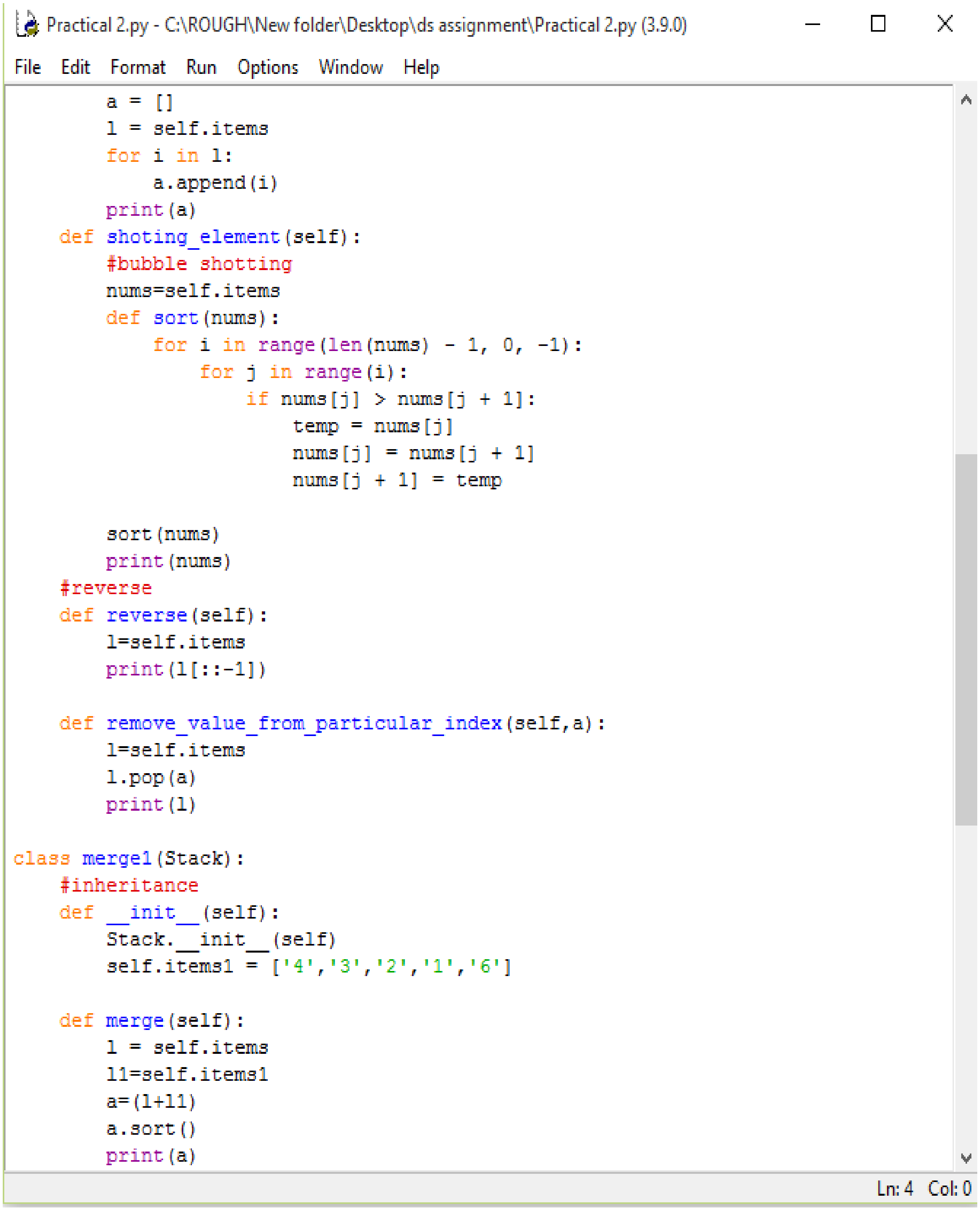
AIM:-2 Implement Linked List. Include options for Insertion, deletion and search of a number, reverse the list and concatenate the lists.

Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-2

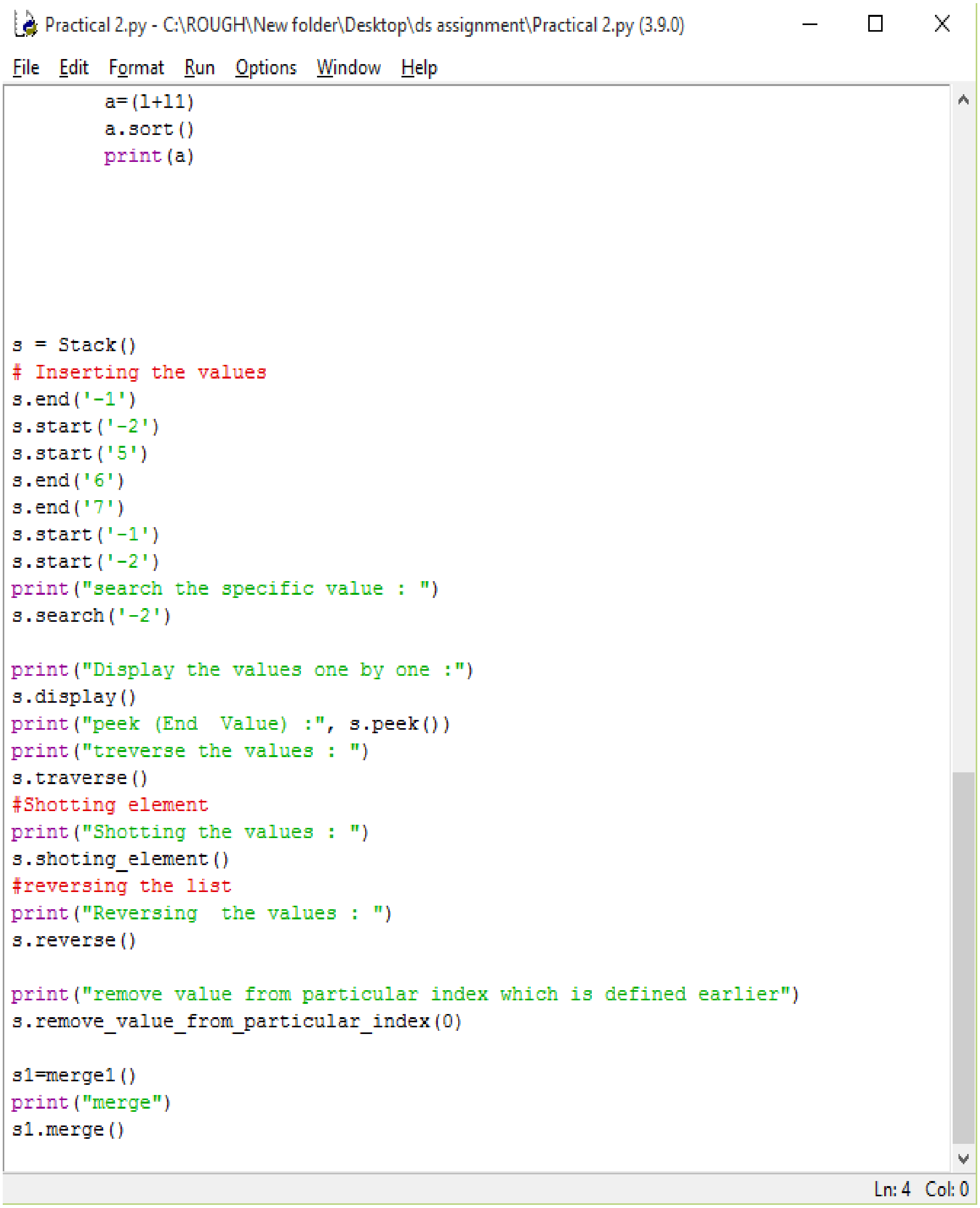
CODE:-


Practical 2.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical 2.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 4 Col: 0

```
# Name  :Hardik D Vaghasiya
# Rollno:4044
# Class :SYCS
class Stack():
    def __init__(self):
        self.items = ['4','3','2','1','Hardik']

    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):
```

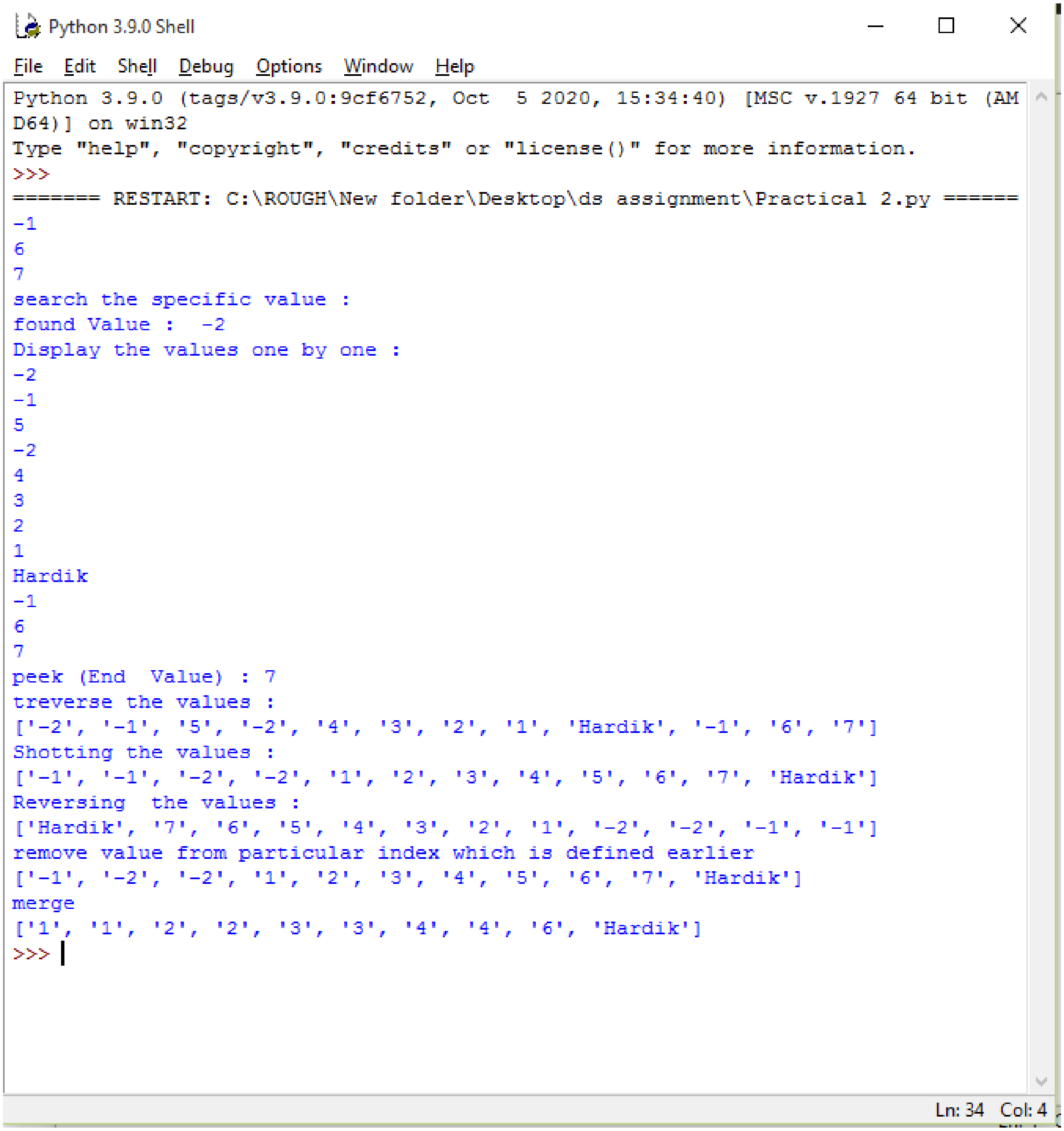

Practical 2.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical 2.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 4 Col: 0

```
        a = []
        l = self.items
        for i in l:
            a.append(i)
        print(a)
    def shoting_element(self):
        #bubble shotting
        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)
```


Practical 2.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical 2.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 4 Col: 0

```
        a=(1+11)
        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:-



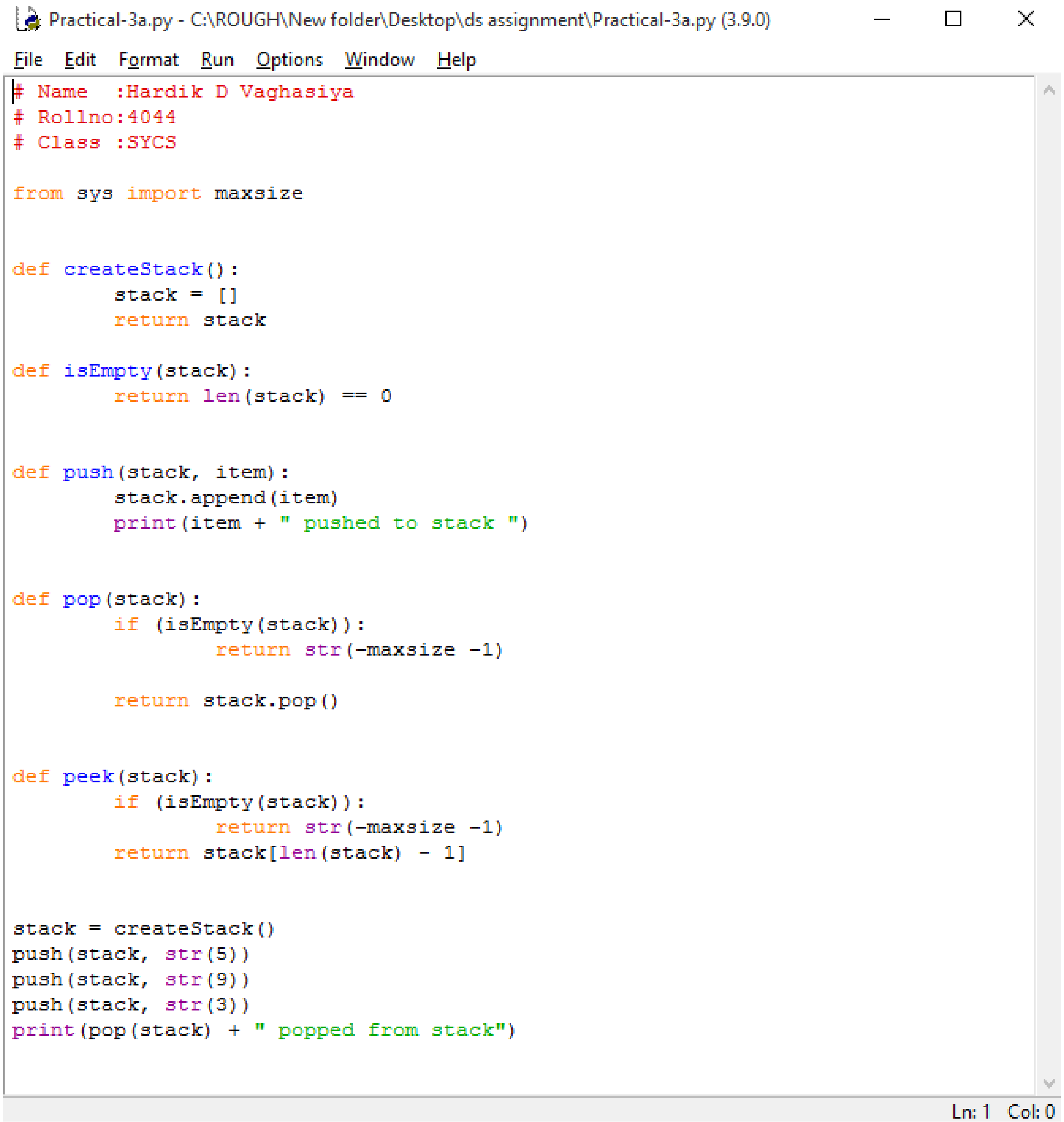
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======= RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical 2.py ======
-1
6
7
search the specific value :
found Value :  -2
Display the values one by one :
-2
-1
5
-2
4
3
2
1
Hardik
-1
6
7
peek (End  Value) : 7
treverse the values :
['-2', '-1', '5', '-2', '4', '3', '2', '1', 'Hardik', '-1', '6', '7']
Shotting the values :
['-1', '-1', '-2', '-2', '1', '2', '3', '4', '5', '6', '7', 'Hardik']
Reversing  the values :
['Hardik', '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', 'Hardik']
merge
['1', '1', '2', '2', '3', '3', '4', '4', '6', 'Hardik']
>>> 
```

PRACTICAL NO:-3A

AIM:-3(A) Perform Stack Operations using Array Implementation.

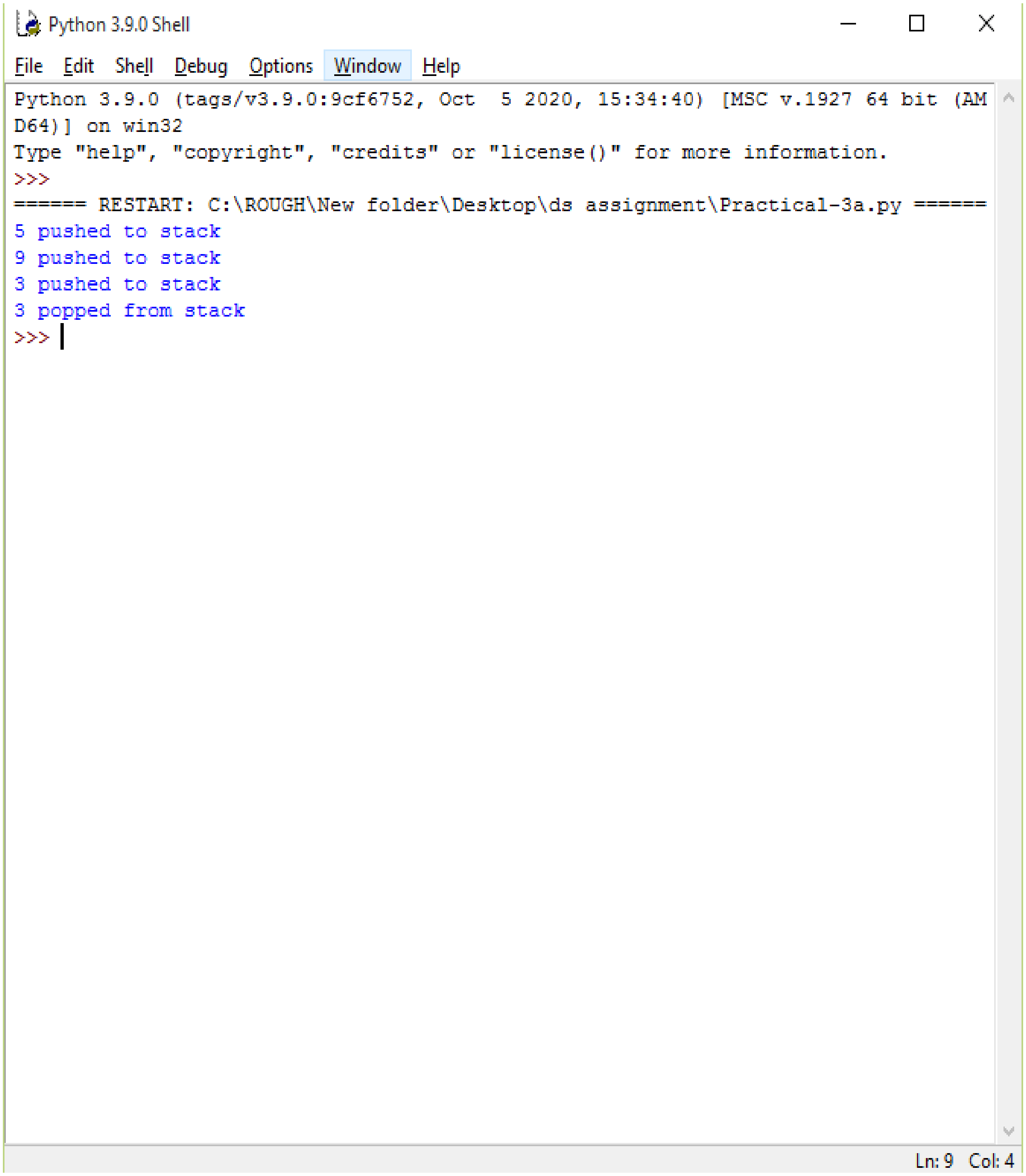
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-3a

CODE:-


Practical-3a.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-3a.py (3.9.0)
File
Edit
Format
Run
Options
Window
Help
Ln: 1
Col: 0

```
# Name  :Hardik D Vaghasiya
# Rollno:4044
# Class :SYCS

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(9))
push(stack, str(3))
print(pop(stack) + " popped from stack")
```

OutPut:-



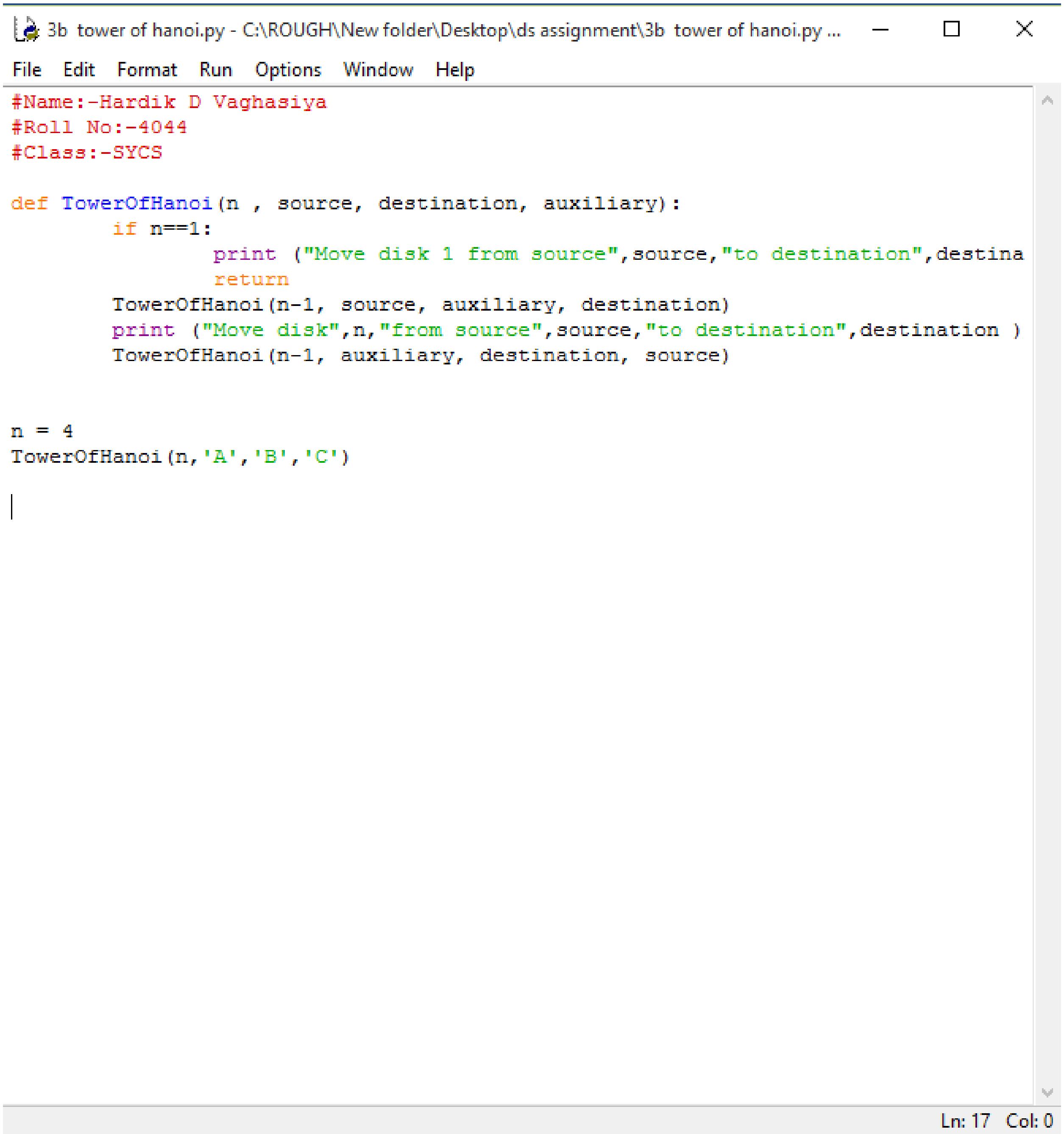
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
====== RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-3a.py ======
5 pushed to stack
9 pushed to stack
3 pushed to stack
3 popped from stack
>>> 
```

PRACTICAL NO:-3B

AIM:-3(B) Implement Tower of Hanoi

Link:- **https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-3b**

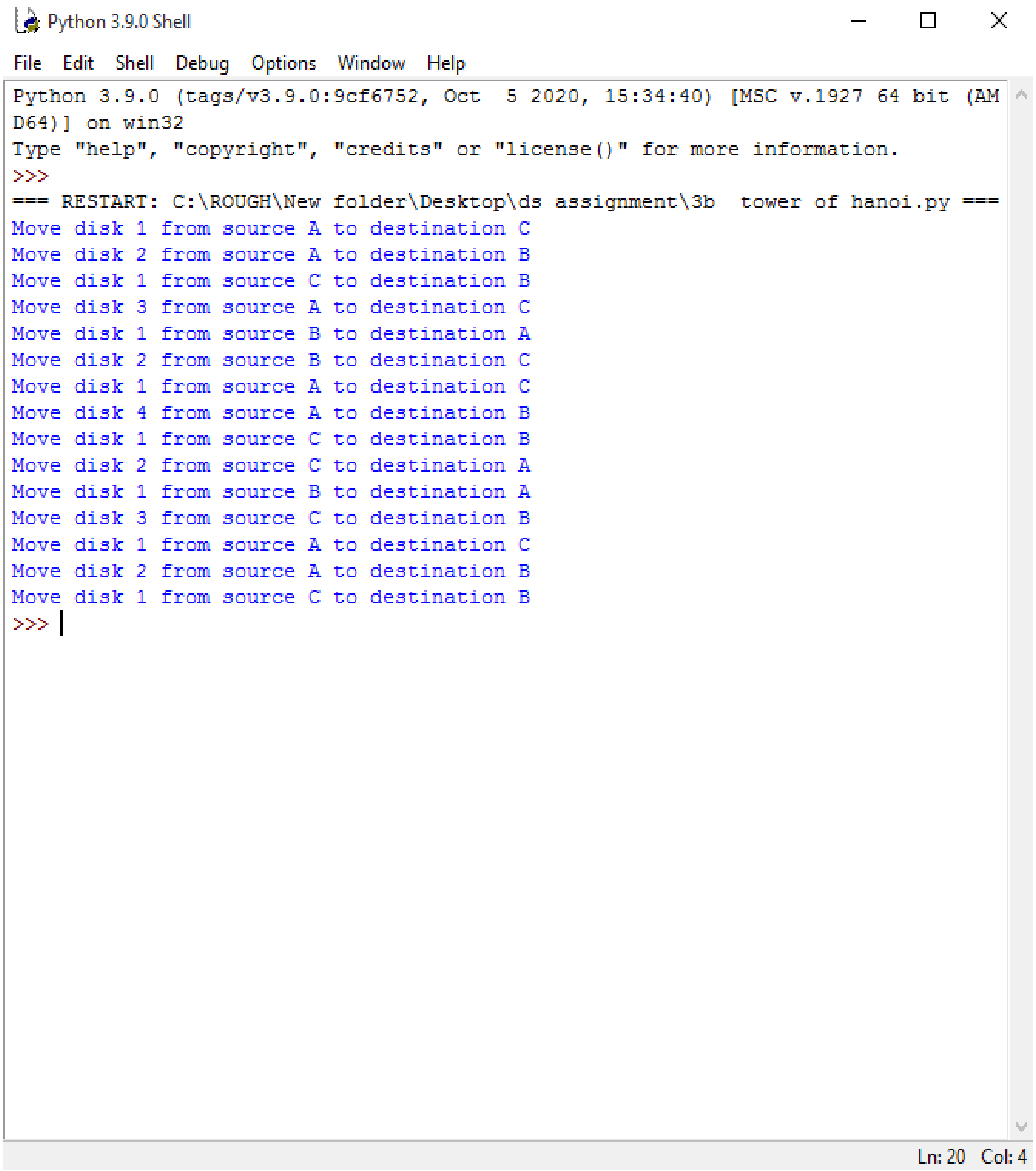
CODE:-



3b tower of hanoi.py - C:\ROUGH\New folder\Desktop\ds assignment\3b tower of hanoi.py ...
File Edit Format Run Options Window Help
Ln: 17 Col: 0

```
#Name:-Hardik D Vaghasiya
#Roll No:-4044
#Class:-SYCS

def TowerOfHanoi(n , source, destination, auxiliary):
        if n==1:
                print ("Move disk 1 from source",source,"to destination",destina
                return
        TowerOfHanoi(n-1, source, auxiliary, destination)
        print ("Move disk",n,"from source",source,"to destination",destination )
        TowerOfHanoi(n-1, auxiliary, destination, source)


n = 4
TowerOfHanoi(n,'A','B','C')
```

OutPut:-



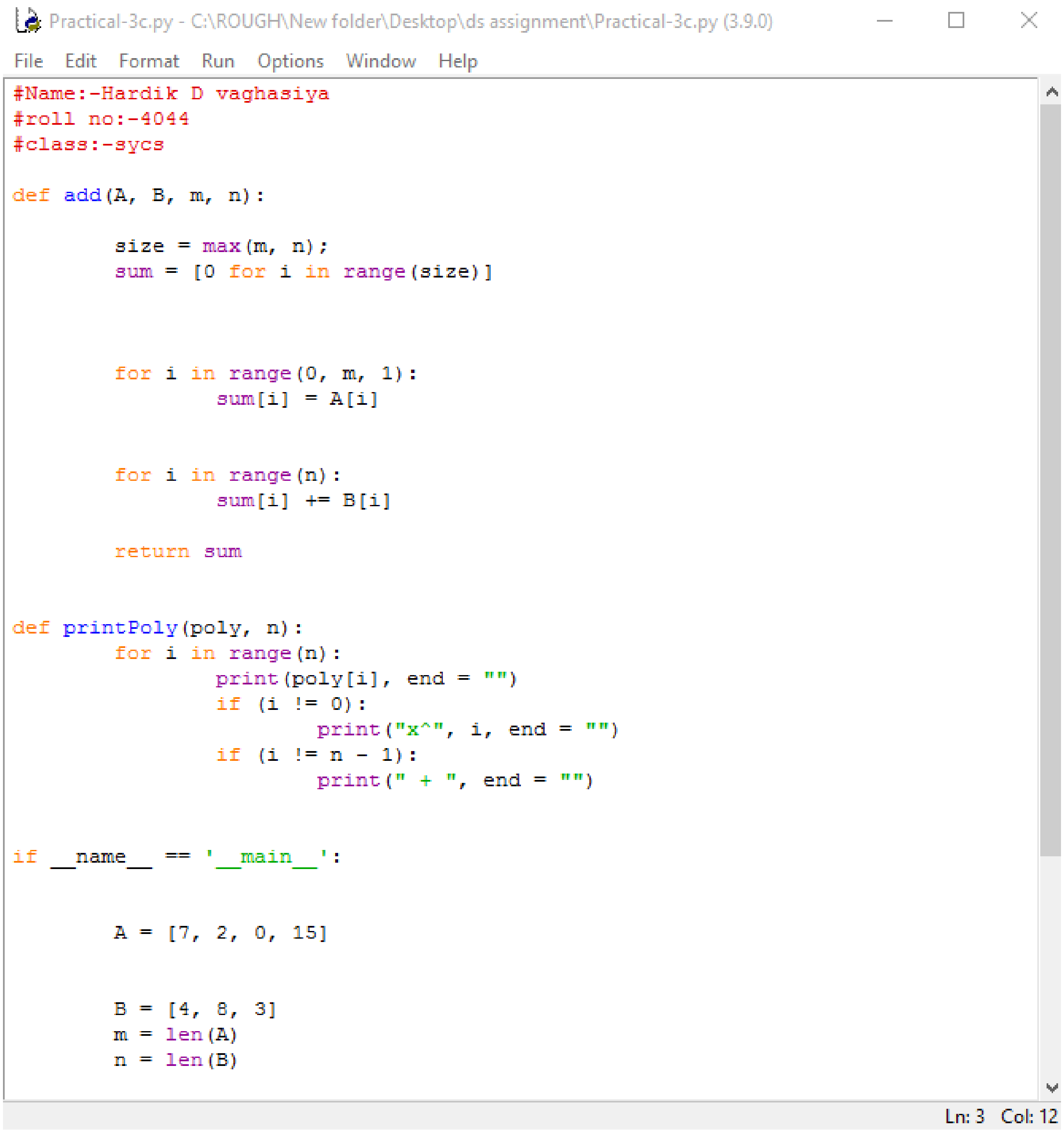
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\ROUGH\New folder\Desktop\ds assignment\3b  tower of hanoi.py ===
Move disk 1 from source A to destination C
Move disk 2 from source A to destination B
Move disk 1 from source C to destination B
Move disk 3 from source A to destination C
Move disk 1 from source B to destination A
Move disk 2 from source B to destination C
Move disk 1 from source A to destination C
Move disk 4 from source A to destination B
Move disk 1 from source C to destination B
Move disk 2 from source C to destination A
Move disk 1 from source B to destination A
Move disk 3 from source C to destination B
Move disk 1 from source A to destination C
Move disk 2 from source A to destination B
Move disk 1 from source C to destination B
>>> 
```

PRACTICAL NO:-3C

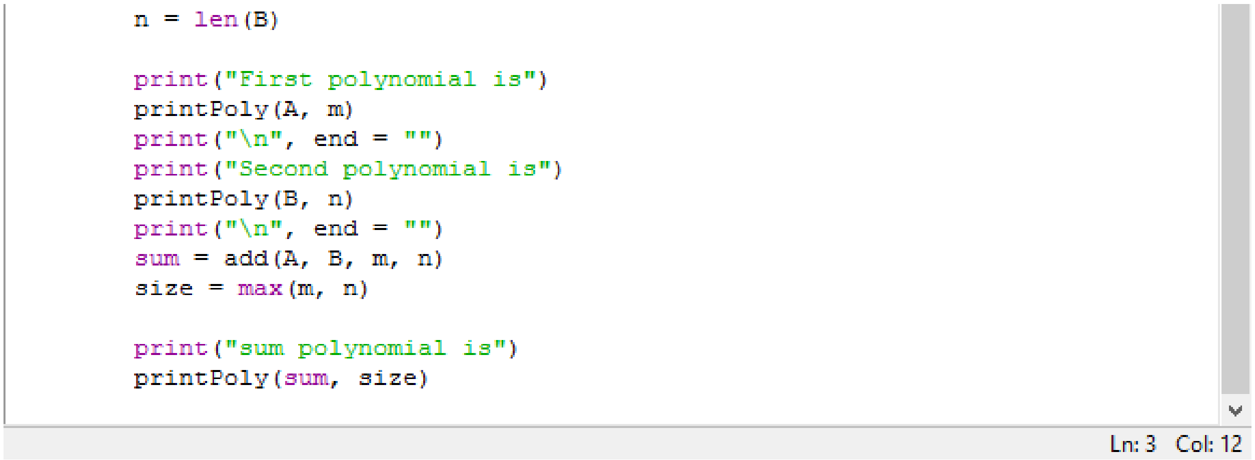
AIM:- WAP to scan a polynomial using linked list and add two polynomial.

Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-3c

CODE:-

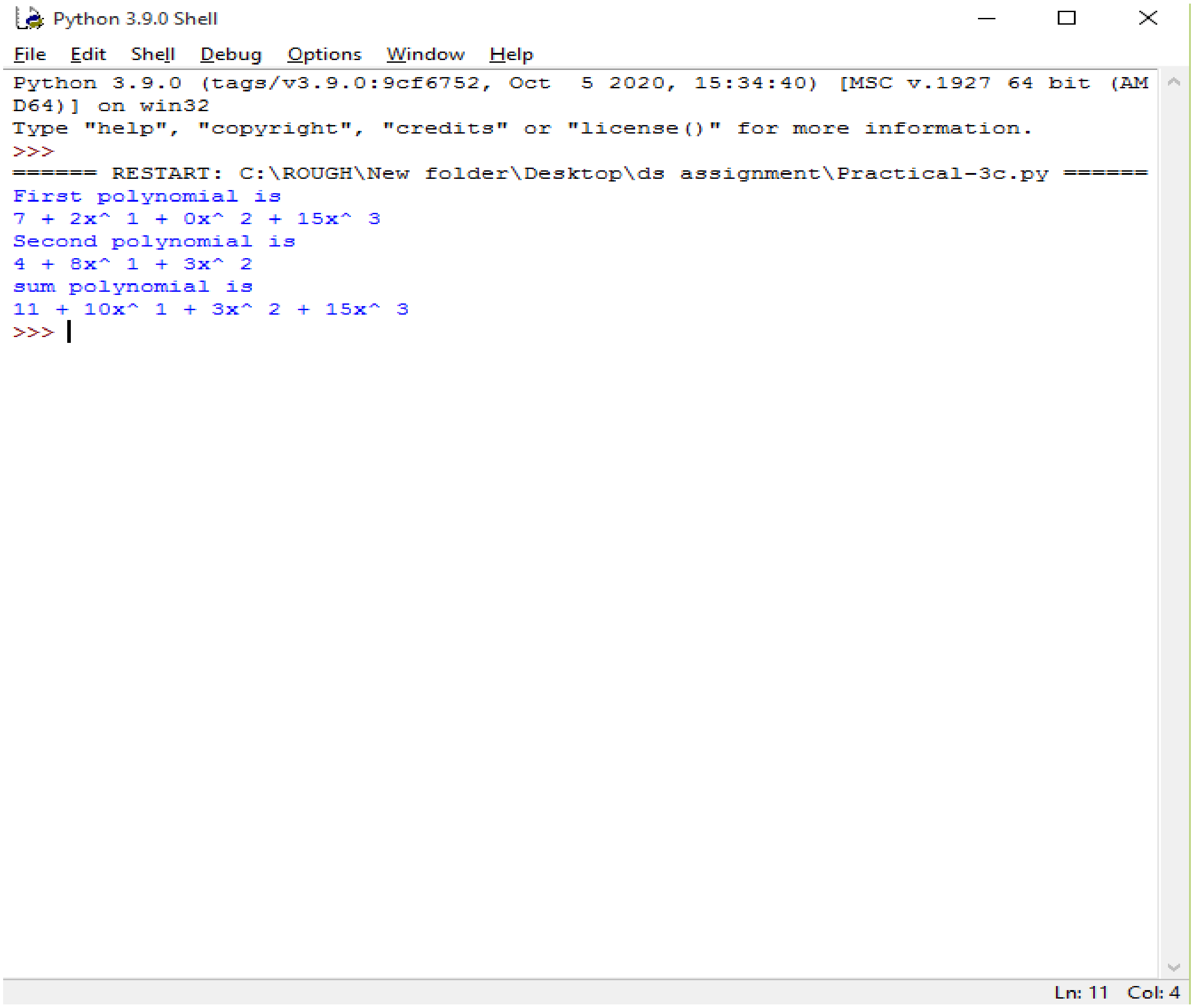

Practical-3c.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-3c.py (3.9.0)
File
Edit
Format
Run
Options
Window
Help
Ln: 3
Col: 12

```
#Name:-Hardik D vaghasiya
#roll no:-4044
#class:-sycs

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 = [7, 2, 0, 15]


        B = [4, 8, 3]
        m = len(A)
        n = len(B)
```



```
    n = len(B)

    print("First polynomial is")
    printPoly(A, m)
    print("\n", end = "")
    print("Second polynomial is")
    printPoly(B, n)
    print("\n", end = "")
    sum = add(A, B, m, n)
    size = max(m, n)

    print("sum polynomial is")
    printPoly(sum, size)
```

OutPut:-



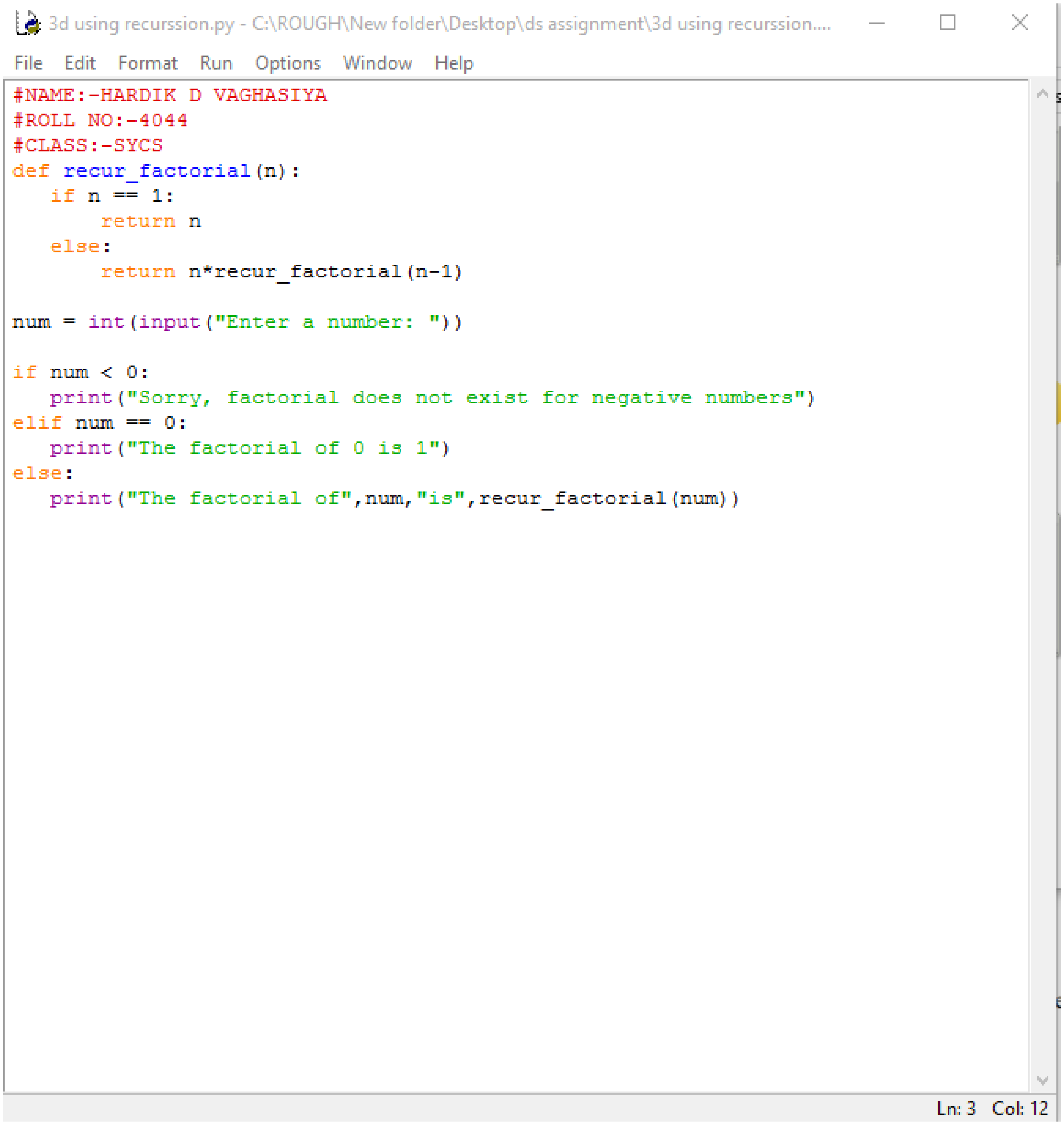
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
====== RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-3c.py ======
First polynomial is
7 + 2x^ 1 + 0x^ 2 + 15x^ 3
Second polynomial is
4 + 8x^ 1 + 3x^ 2
sum polynomial is
11 + 10x^ 1 + 3x^ 2 + 15x^ 3
>>> 
```

PRACTICAL NO:-3D

AIM:-3(D)WAP to calculate factorial and to compute the factors of a given no. (i) using recursion, (ii) using iteration.

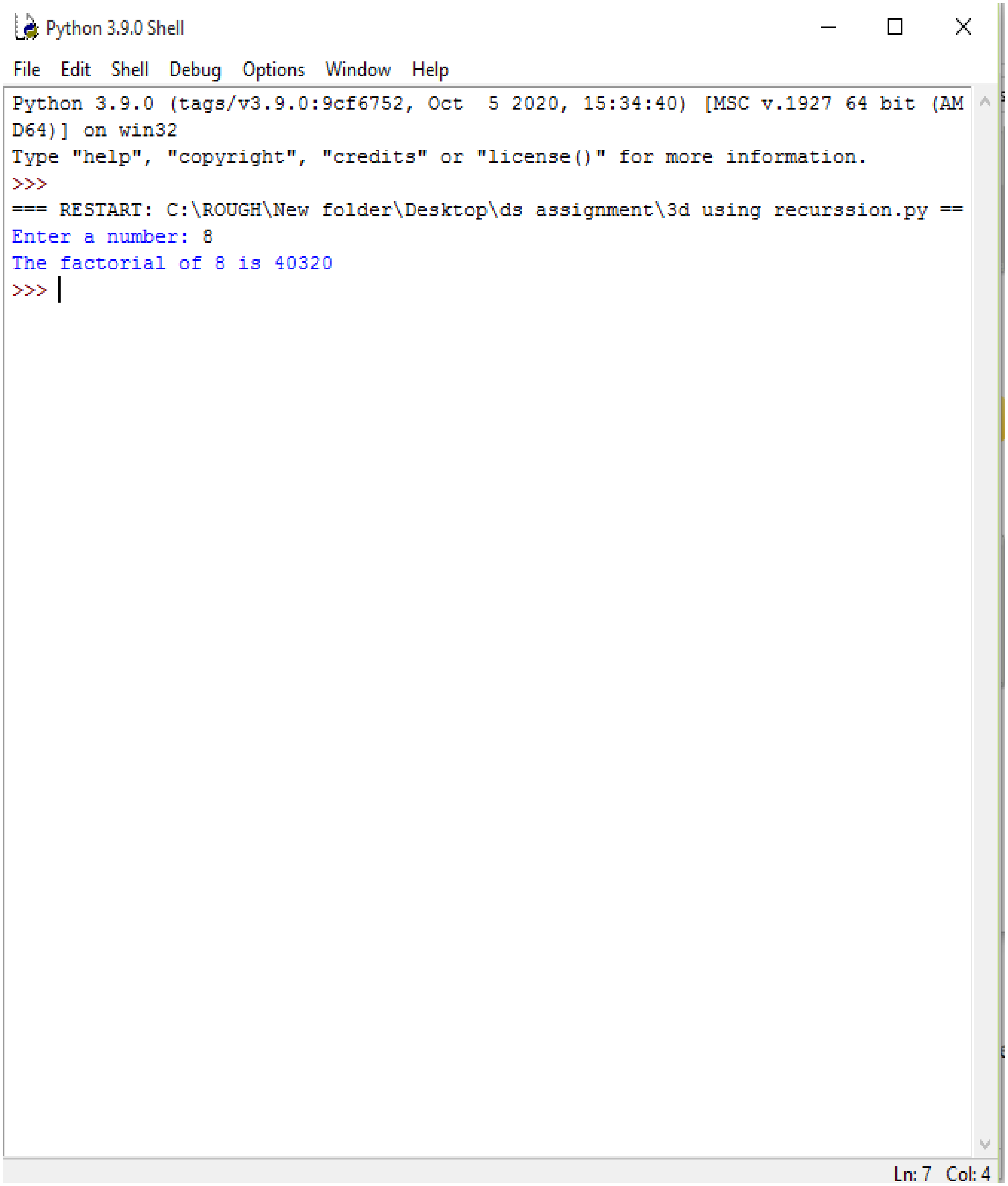
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-3d

CODE:- (i) using recursion


3d using recurssion.py - C:\ROUGH\New folder\Desktop\ds assignment\3d using recurssion....
File Edit Format Run Options Window Help
Ln: 3 Col: 12

```
#NAME:-HARDIK D VAGHASIYA
#ROLL NO:-4044
#CLASS:-SYCS
def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n*recur_factorial(n-1)

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

if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of",num,"is",recur_factorial(num))
```

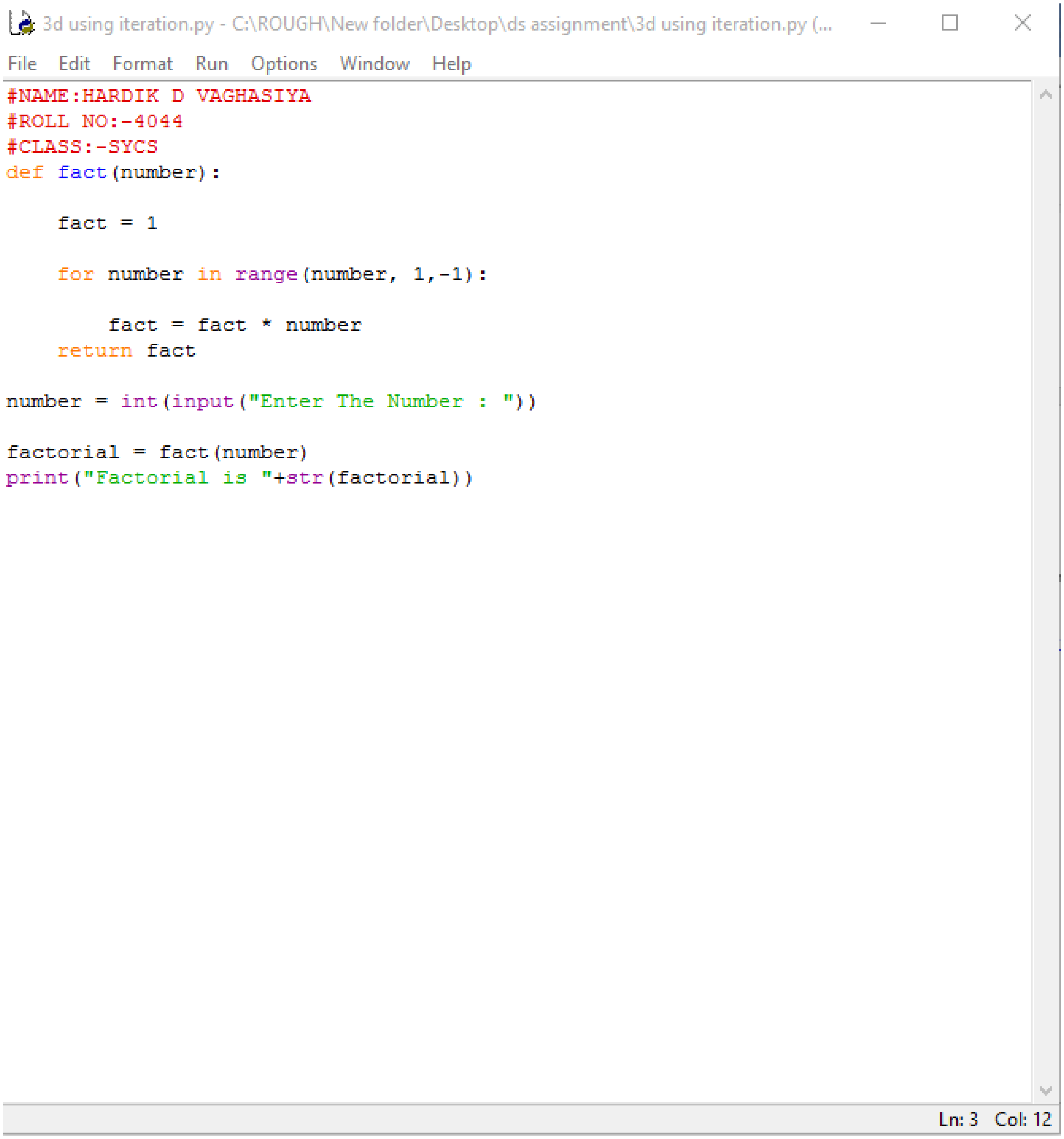
OutPut:-



```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=== RESTART: C:\ROUGH\New folder\Desktop\ds assignment\3d using recurssion.py ==
Enter a number: 8
The factorial of 8 is 40320
>>> 
```

(ii) using iteration.

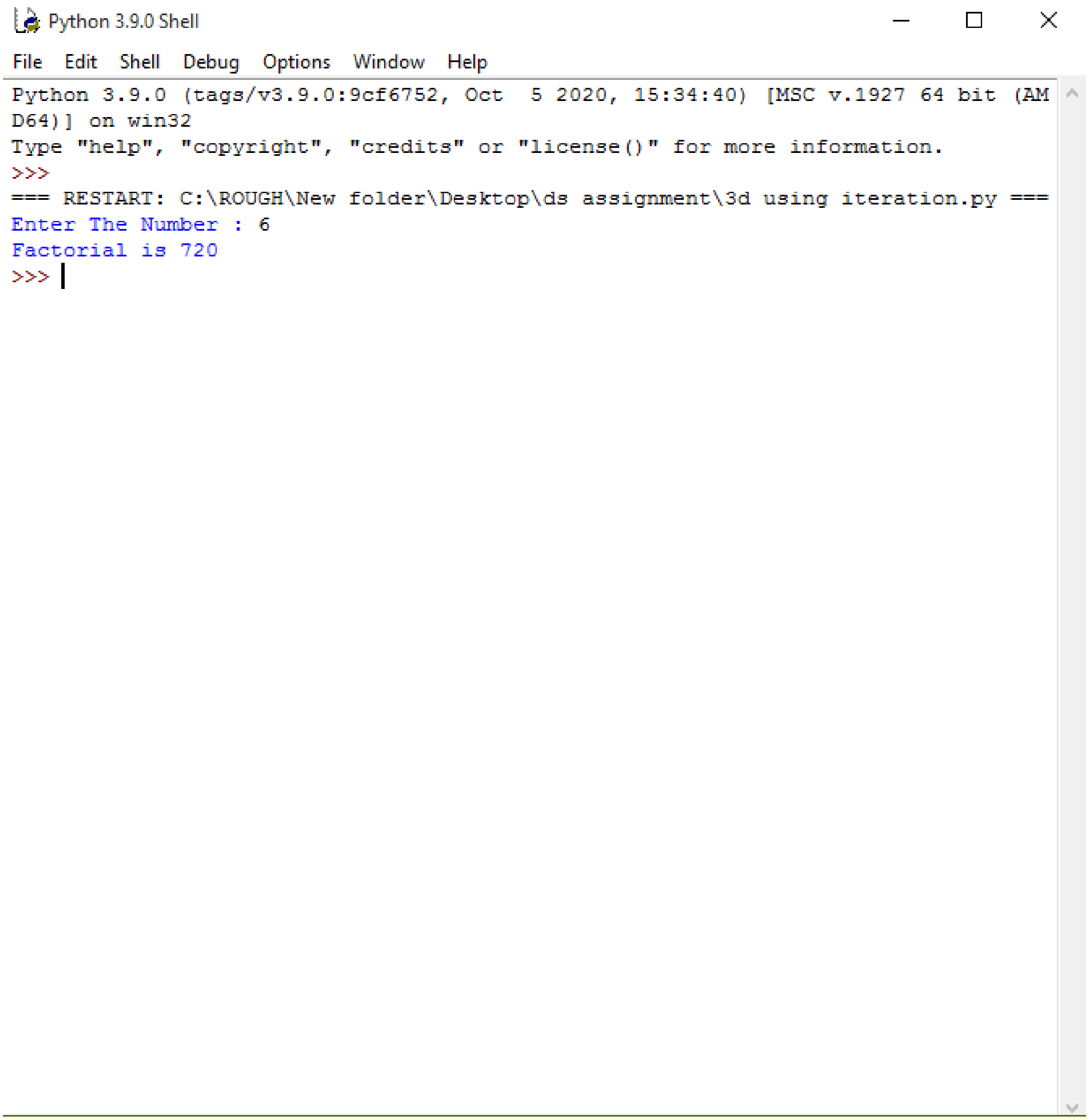
CODE:-



3d using iteration.py - C:\ROUGH\New folder\Desktop\ds assignment\3d using iteration.py (...
File Edit Format Run Options Window Help
Ln: 3 Col: 12

```
#NAME:HARDIK D VAGHASIYA
#ROLL NO:-4044
#CLASS:-SYCS
def fact(number):

    fact = 1

    for number in range(number, 1,-1):

        fact = fact * number
    return fact

number = int(input("Enter The Number : "))

factorial = fact(number)
print("Factorial is "+str(factorial))
```

OutPut:-



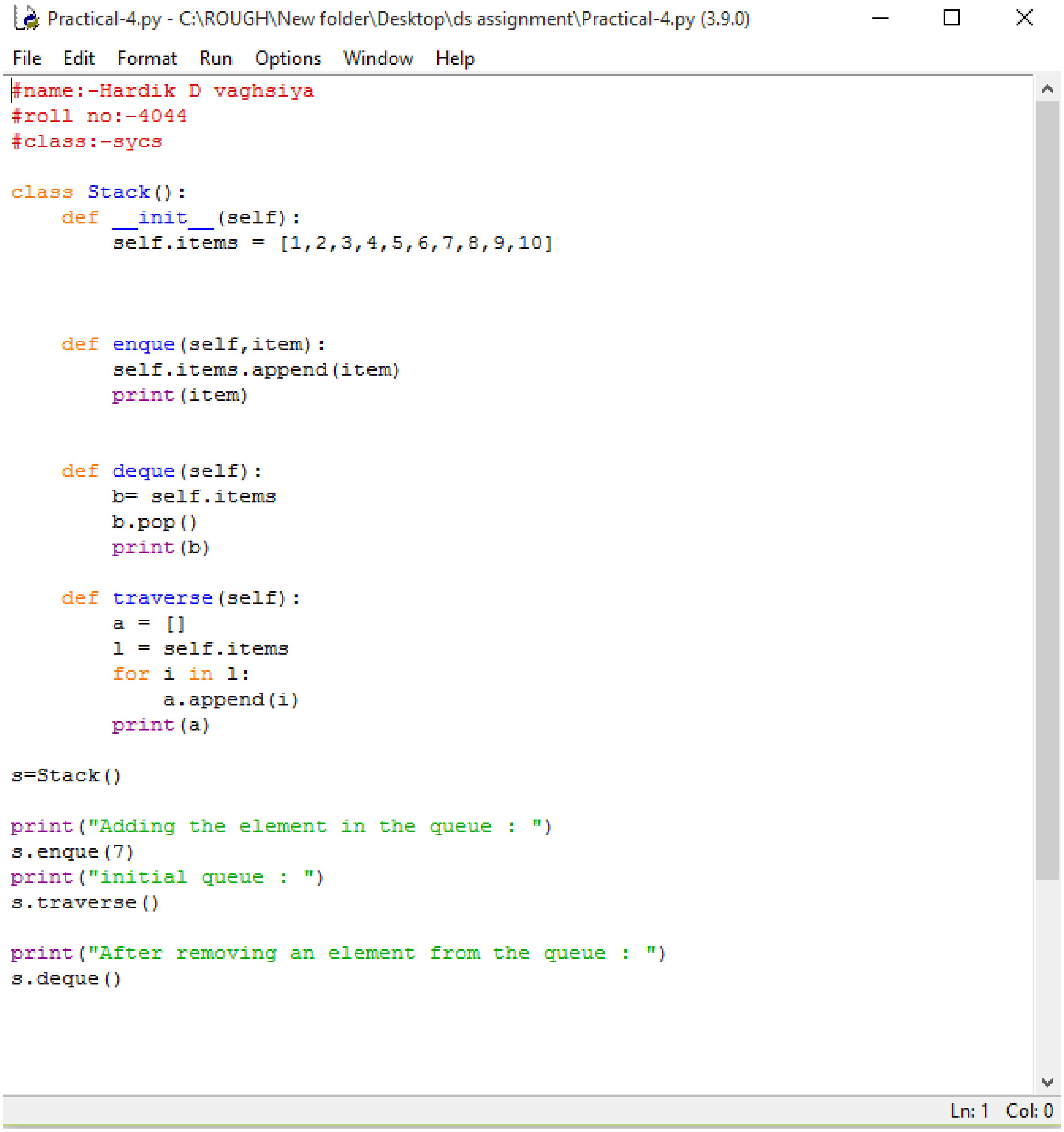
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=== RESTART: C:\ROUGH\New folder\Desktop\ds assignment\3d using iteration.py ===
Enter The Number : 6
Factorial is 720
>>> 
```

PRACTICAL NO:-4

AIM:- Perform Queues operations using Circular Array implementation

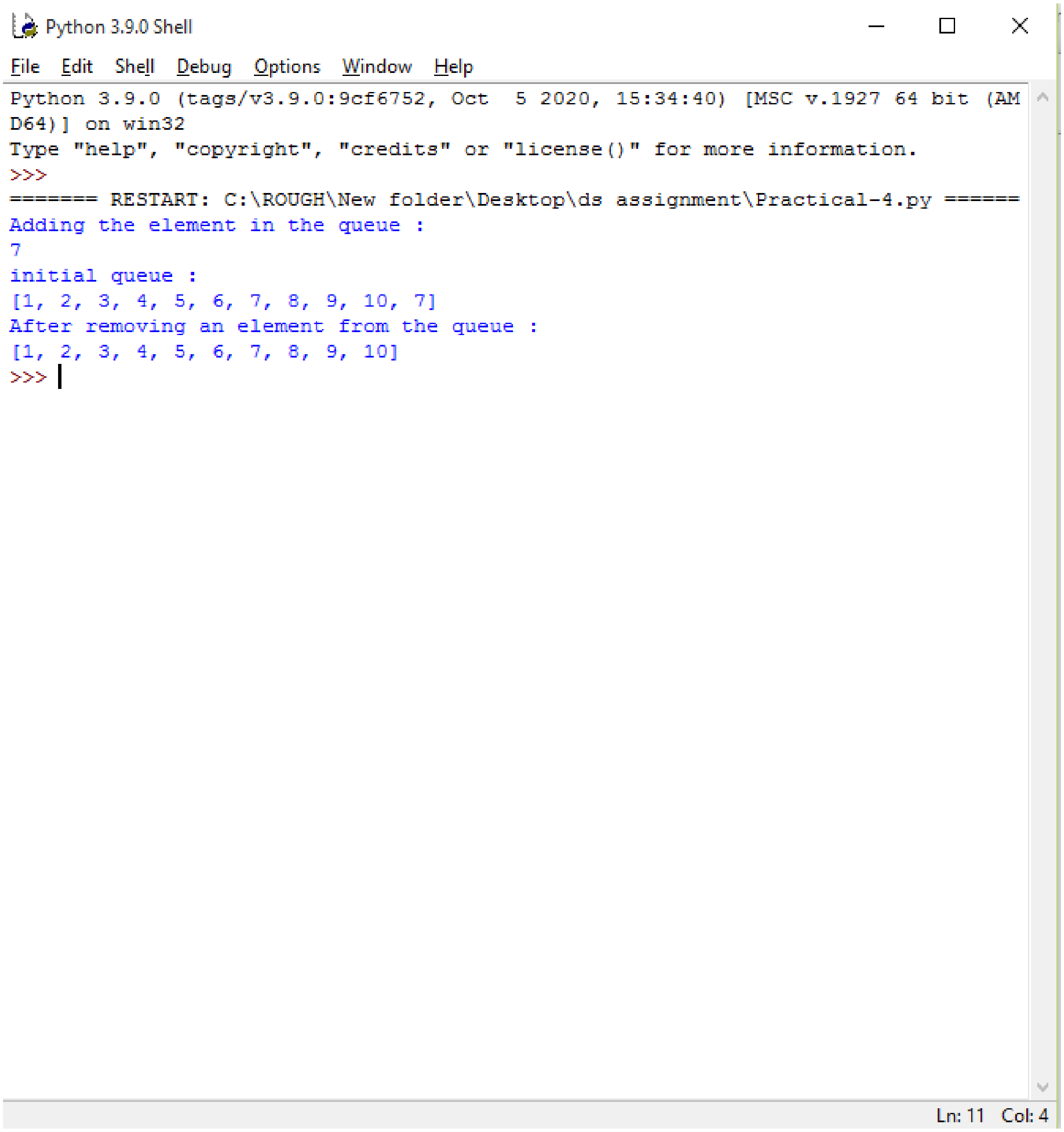
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-4

CODE:-


Practical-4.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-4.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```
#name:-Hardik D vaghsiya
#roll no:-4044
#class:-sycs

class Stack():
    def __init__(self):
        self.items = [1,2,3,4,5,6,7,8,9,10]



    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(7)
print("initial queue : ")
s.traverse()

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

OutPut:-



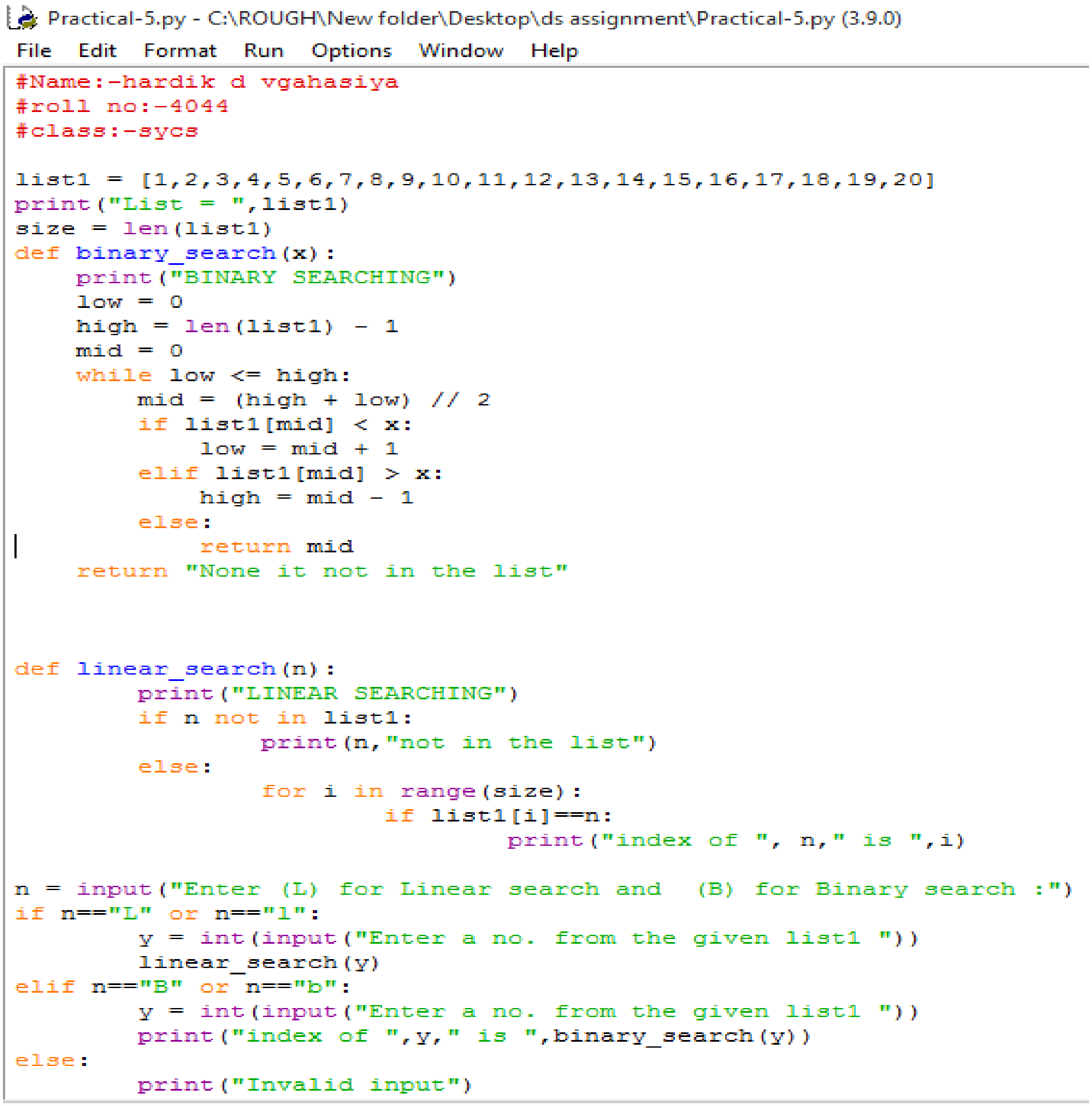
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======= RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-4.py ======
Adding the element in the queue :
7
initial queue :
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 7]
After removing an element from the queue :
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> 
```

PRACTICAL NO:-5

AIM:- Write a program to search an element from a list. Give user the option to perform Linear or Binary search.

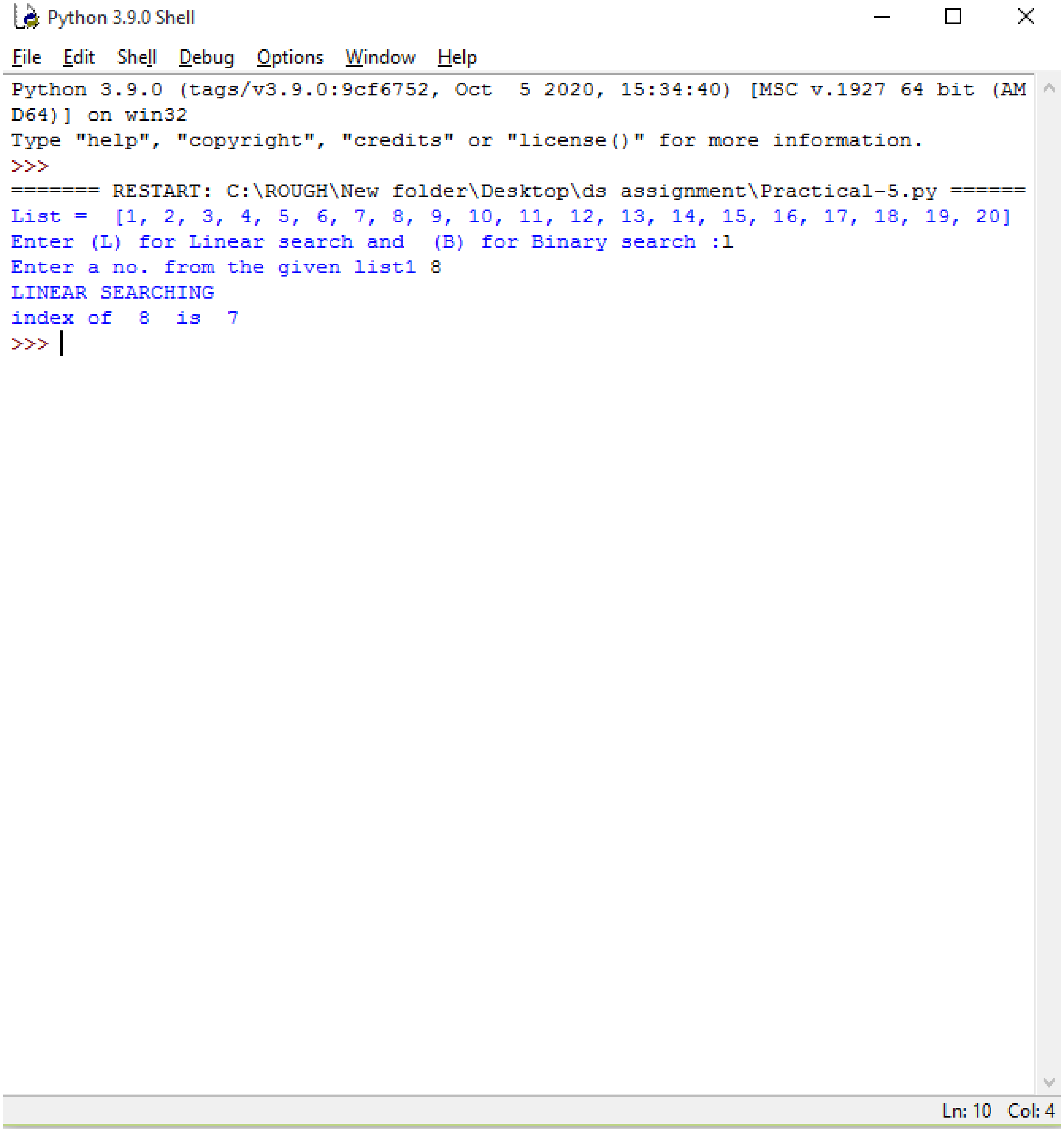
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-5

CODE:-


Practical-5.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-5.py (3.9.0)
File Edit Format Run Options Window Help

```
#Name:-hardik d vgahasiya
#roll no:-4044
#class:-sycs

list1 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
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:-



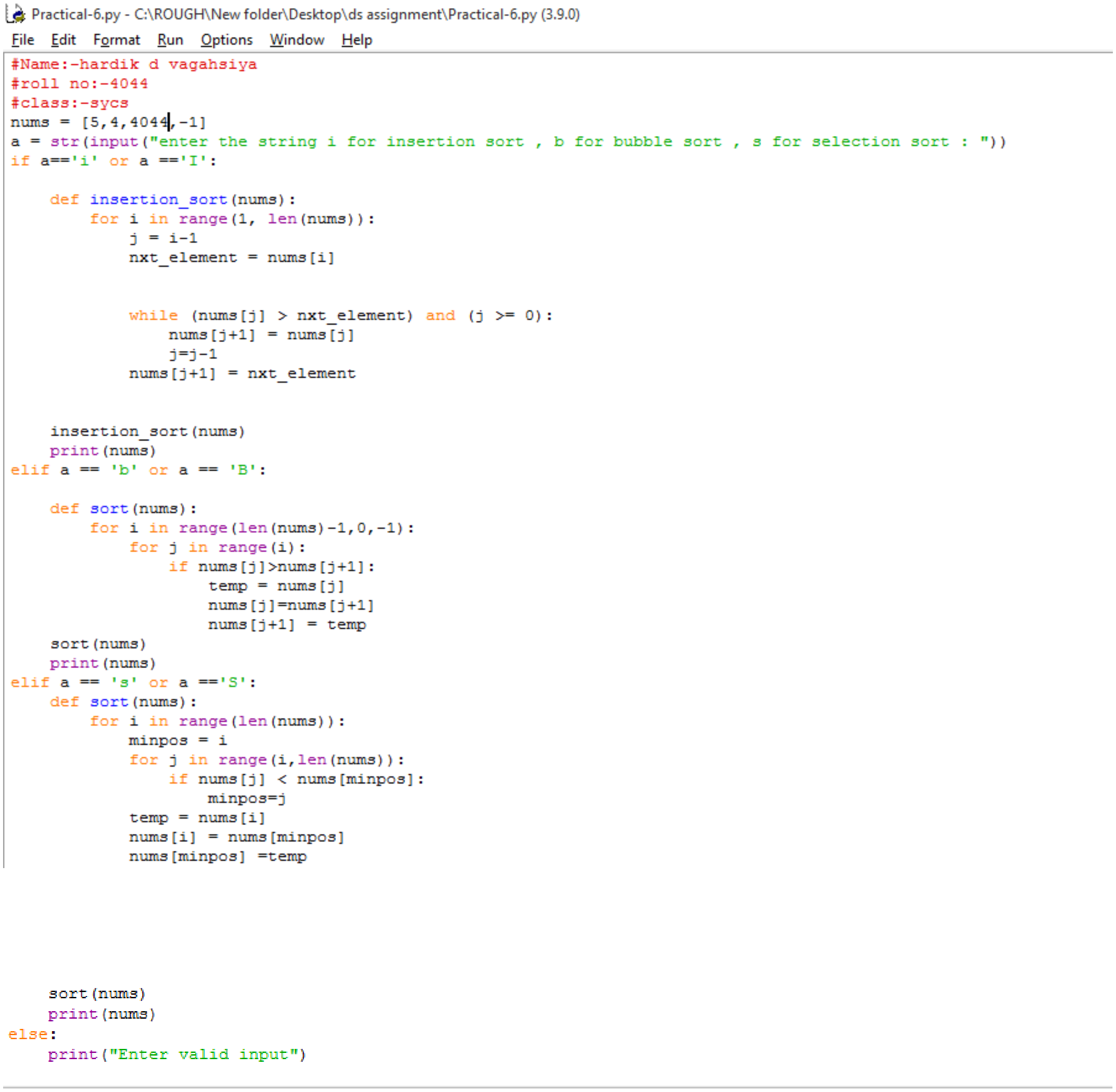
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======= RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-5.py ======
List =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
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
>>> 
```

PRACTICAL NO:-6

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

Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-6

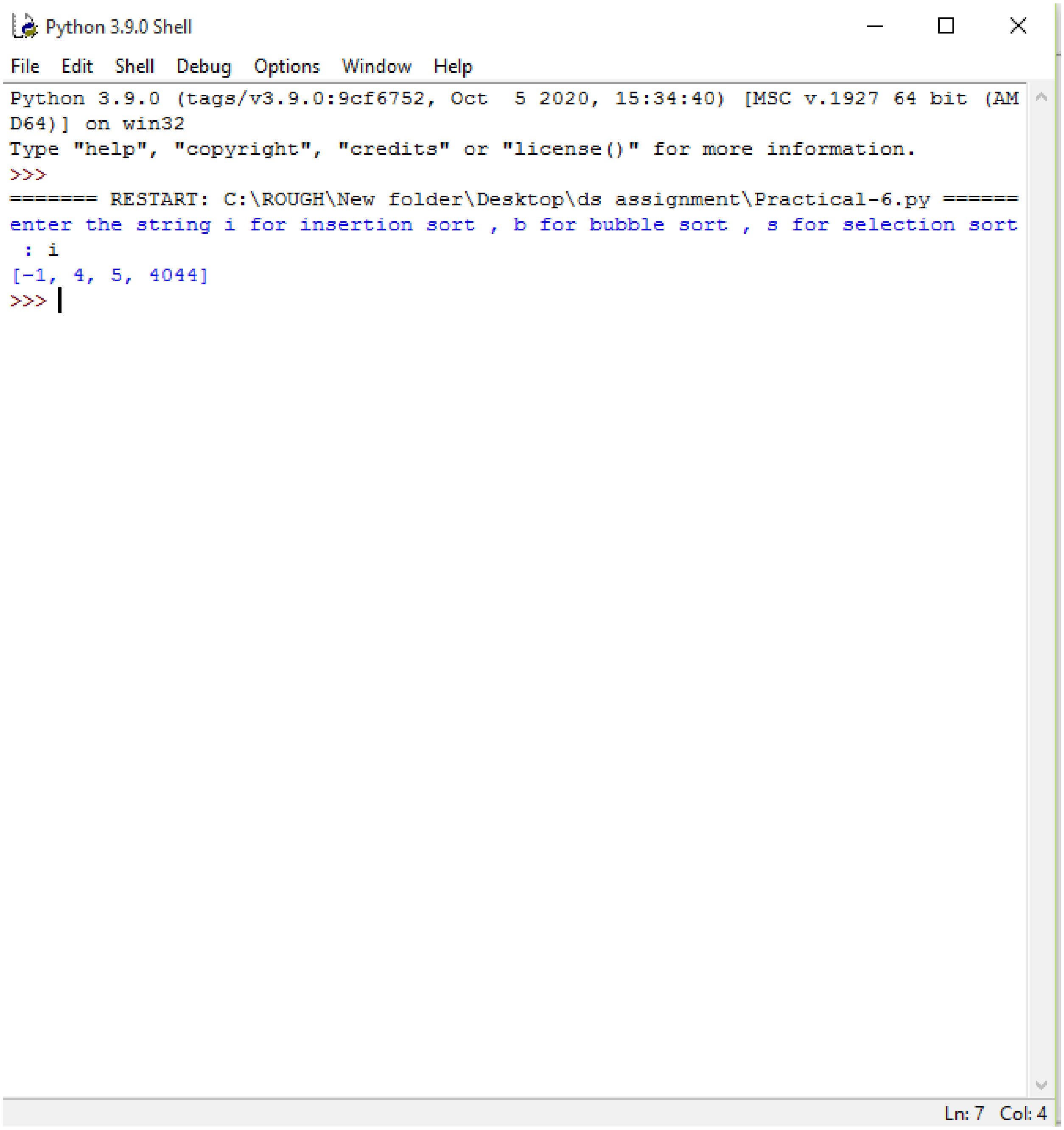
CODE:-



Practical-6.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-6.py (3.9.0)
File Edit Format Run Options Window Help

```
#Name:-hardik d vagahsiya
#roll no:-4044
#class:-sycs
nums = [5,4,4044,-1]
a = str(input("enter the string i for insertion sort , b for bubble sort , s for selection sort : "))
if a=='i' or a =='I':

    def insertion_sort(nums):
        for i in range(1, len(nums)):
            j = i-1
            nxt_element = nums[i]


            while (nums[j] > nxt_element) and (j >= 0):
                nums[j+1] = nums[j]
                j=j-1
            nums[j+1] = nxt_element


    insertion_sort(nums)
    print(nums)
elif a == 'b' or a == 'B':

    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)
elif a == 's' or a =='S':
    def sort(nums):
        for i in range(len(nums)):
            minpos = i
            for j in range(i,len(nums)):
                if nums[j] < nums[minpos]:
                    minpos=j
            temp = nums[i]
            nums[i] = nums[minpos]
            nums[minpos] =temp




    sort(nums)
    print(nums)
else:
    print("Enter valid input")
```

OutPut:-



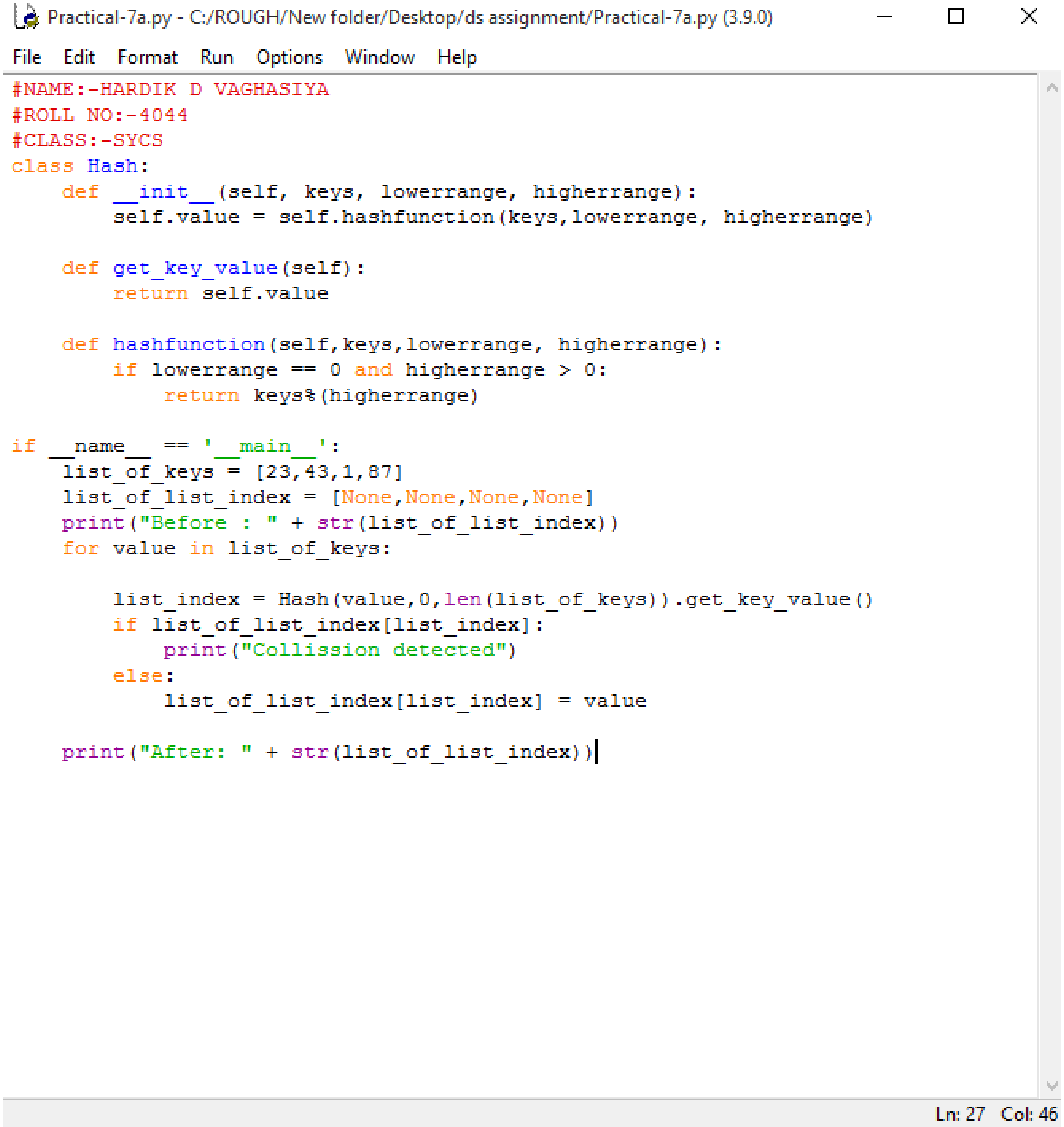
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======= RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-6.py ======
enter the string i for insertion sort , b for bubble sort , s for selection sort
 : i
[-1, 4, 5, 4044]
>>> 
```

PRACTICAL NO:-7

AIM:- Implement the following for Hashing: (a). Write a program to implement the collision technique.(b) Write a program to implement the concept of linear probing.

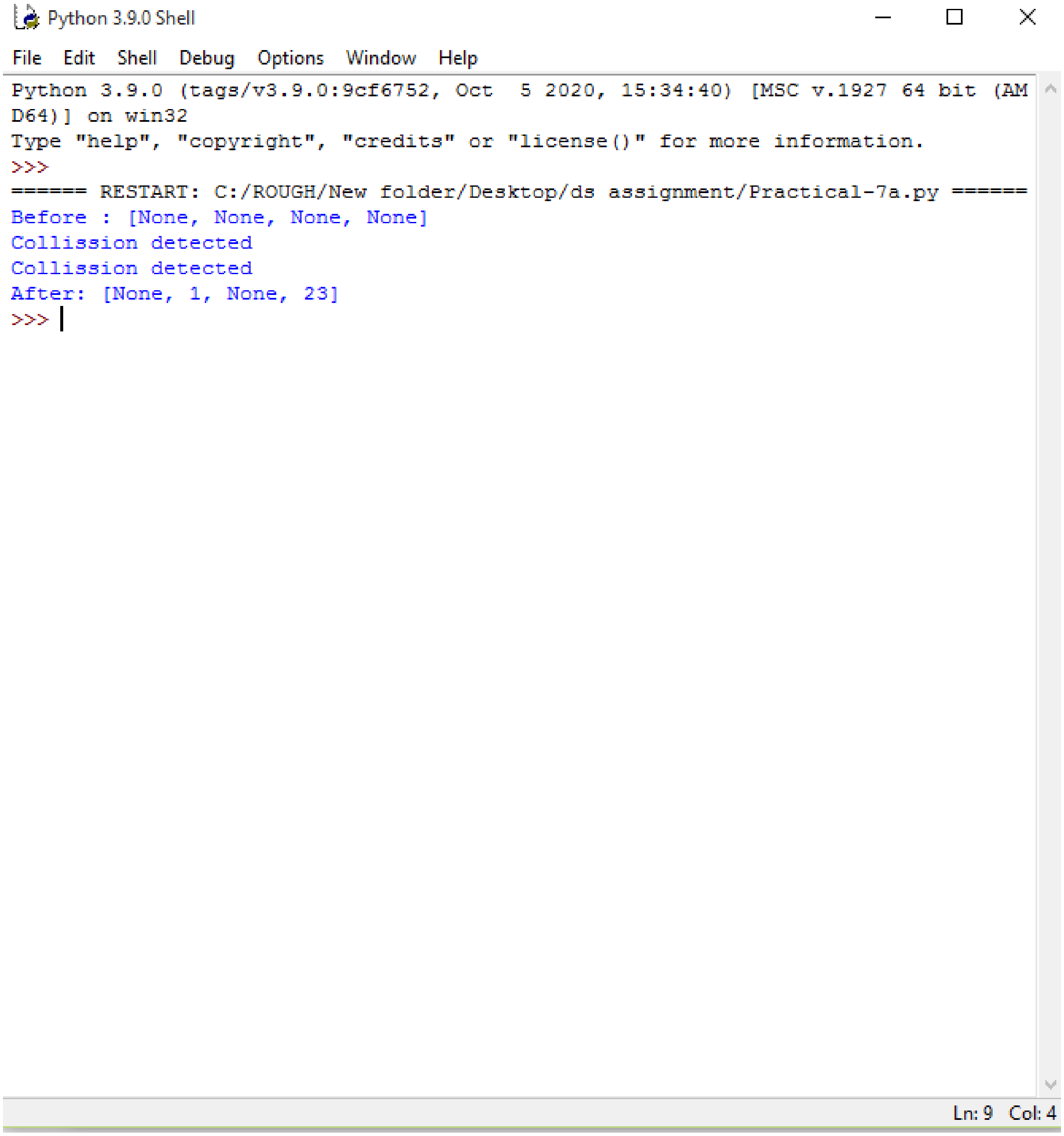
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-7a

CODE:-(a) a


Practical-7a.py - C:/ROUGH/New folder/Desktop/ds assignment/Practical-7a.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 27 Col: 46

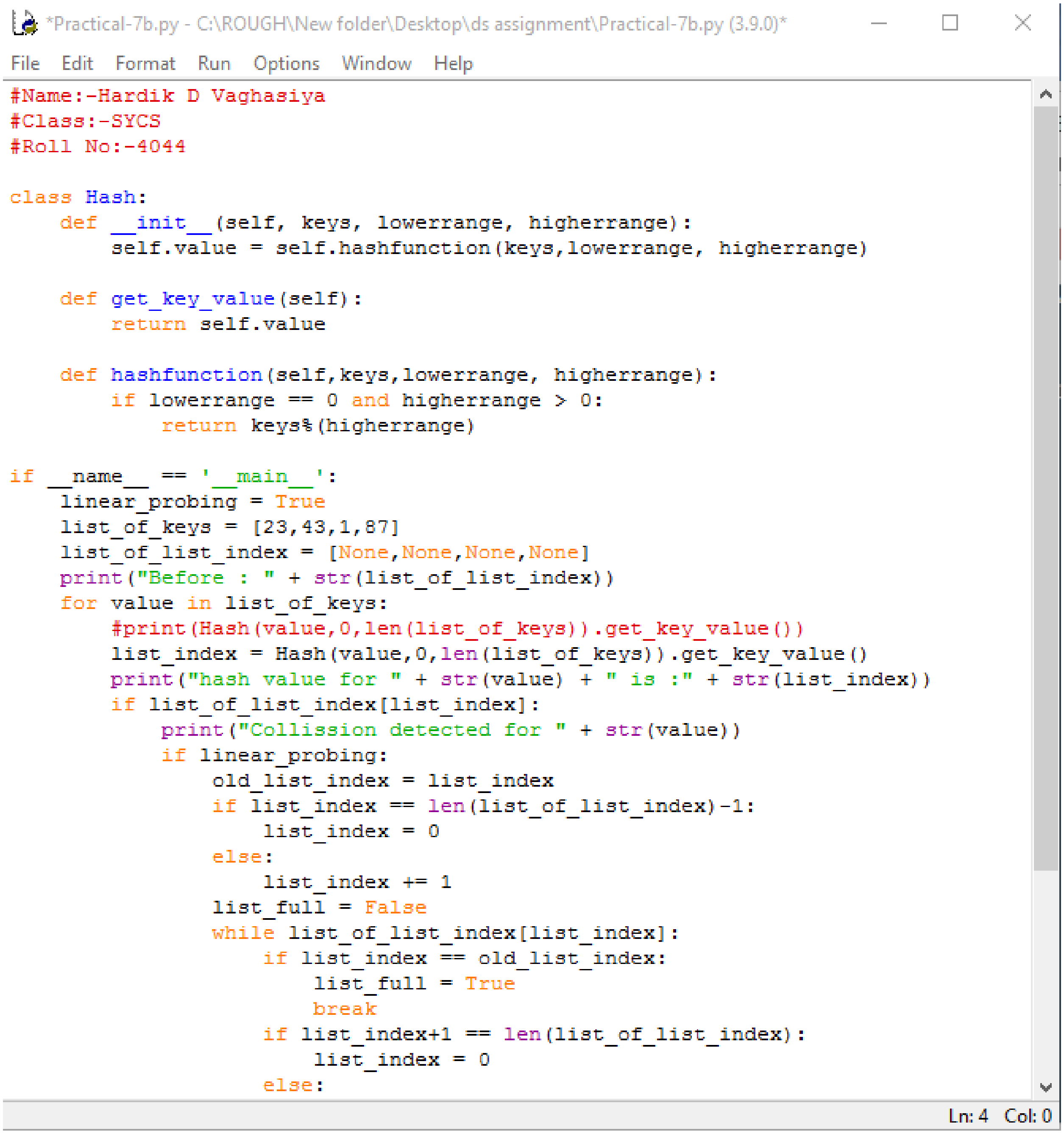
```
#NAME:-HARDIK D VAGHASIYA
#ROLL NO:-4044
#CLASS:-SYCS
class Hash:
    def __init__(self, keys, lowerrange, higherrange):
        self.value = self.hashfunction(keys,lowerrange, higherrange)

    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__':
    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:

        list_index = Hash(value,0,len(list_of_keys)).get_key_value()
        if list_of_list_index[list_index]:
            print("Collission detected")
        else:
            list_of_list_index[list_index] = value

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

OutPut:-

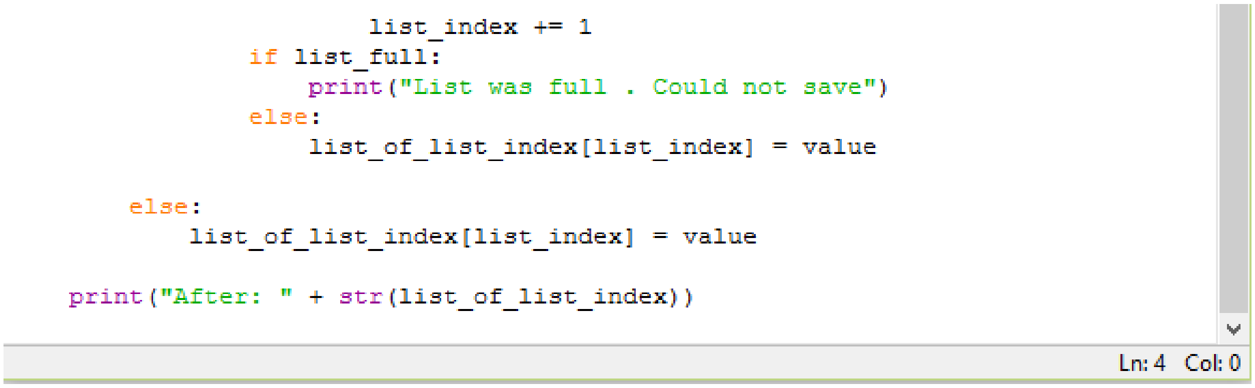


```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: C:/ROUGH/New folder/Desktop/ds assignment/Practical-7a.py ======
Before : [None, None, None, None]
Collission detected
Collission detected
After: [None, 1, None, 23]
>>> 
```

CODE:-(b)

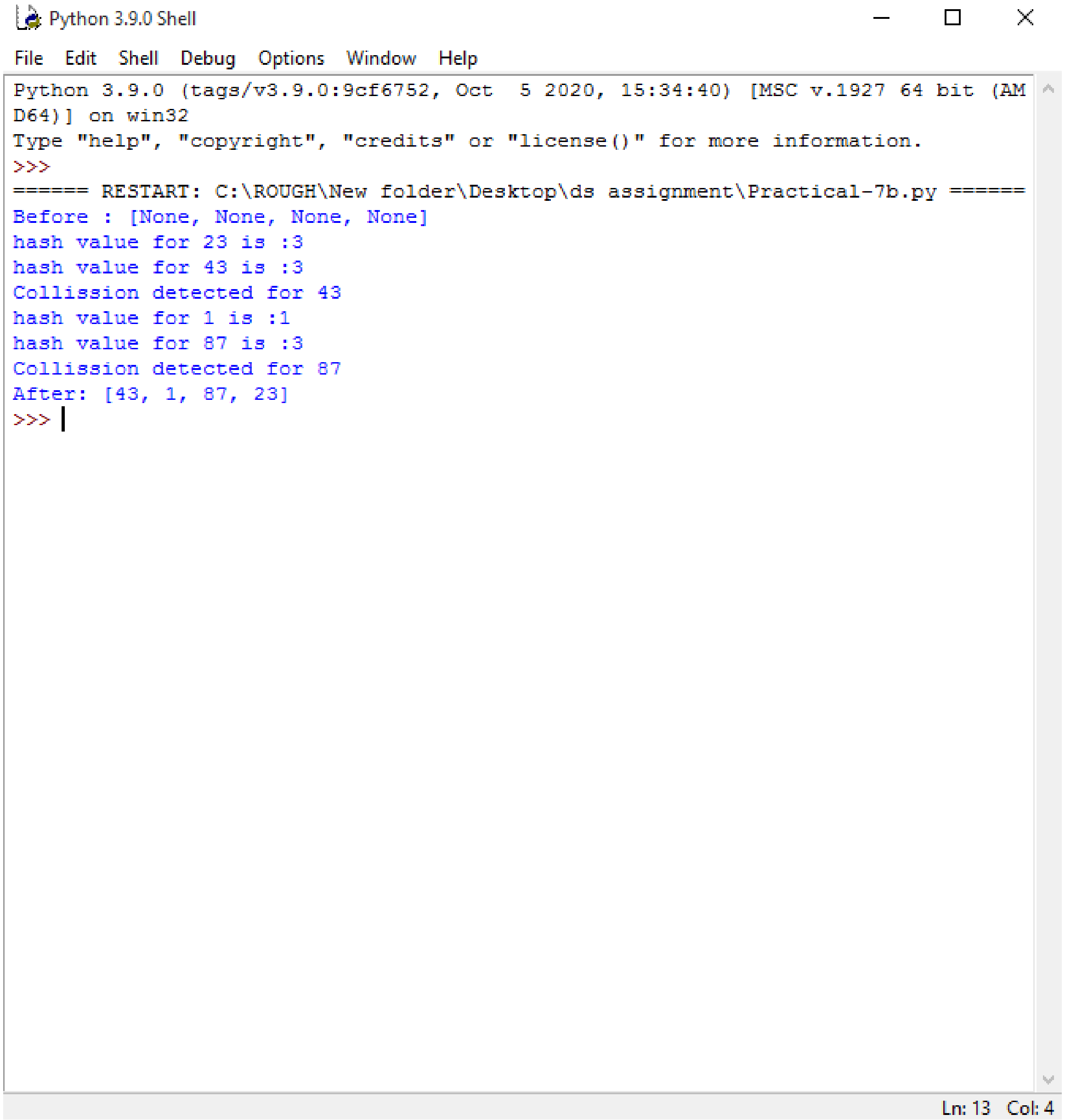

Practical-7b.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-7b.py (3.9.0)
File
Edit
Format
Run
Options
Window
Help
Ln: 4
Col: 0

```
#Name:-Hardik D Vaghasiya
#Class:-SYCS
#Roll No:-4044

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:-(b)



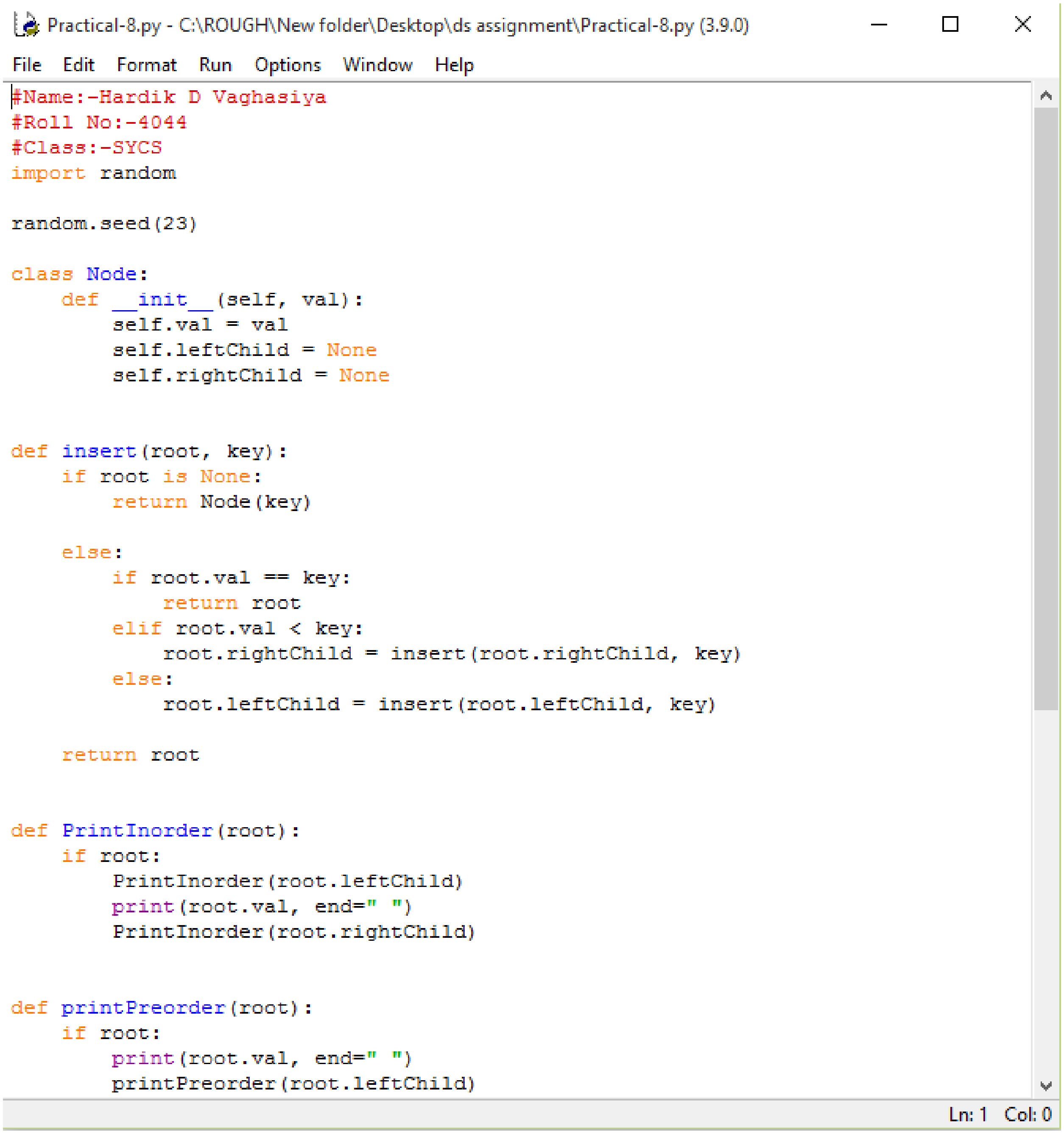
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
====== RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-7b.py ======
Before : [None, None, None, None]
hash value 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:- Write a program for inorder, postorder and preorder traversal of tree.

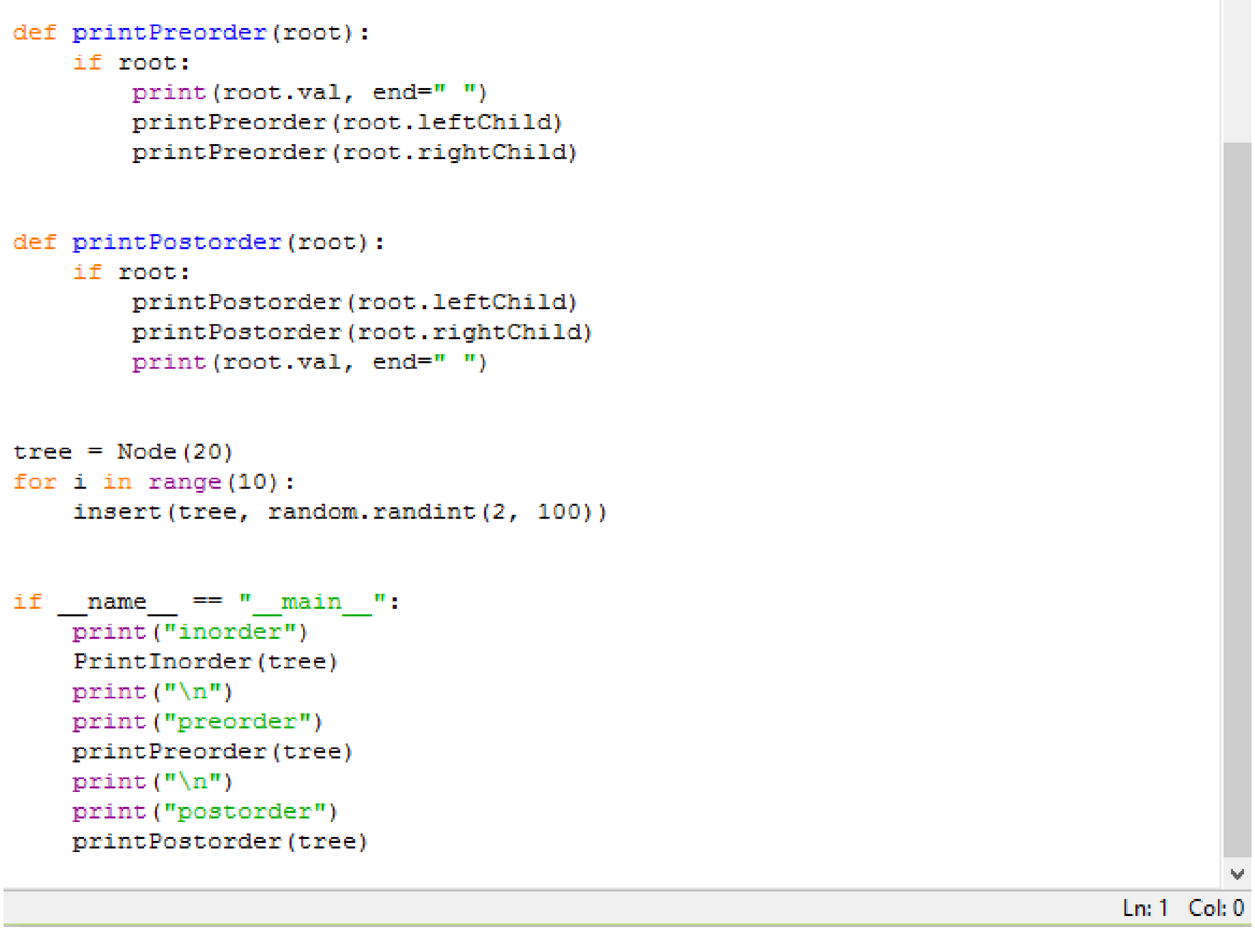
Link:- https://github.com/Hardikvaghasiya/4044-DS/blob/main/Practical-8

CODE:-



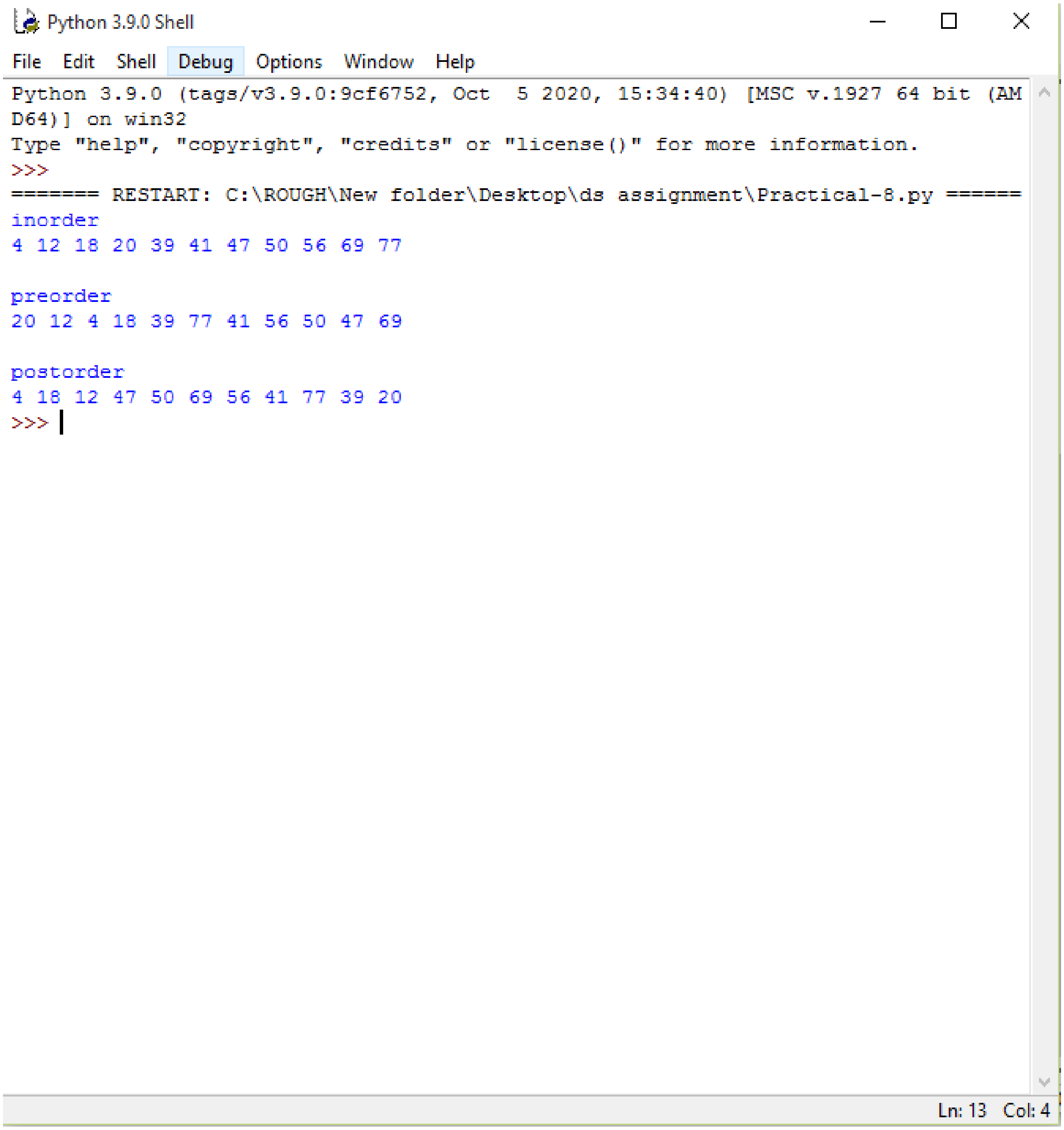
Practical-8.py - C:\ROUGH\New folder\Desktop\ds assignment\Practical-8.py (3.9.0)
File
Edit
Format
Run
Options
Window
Help
Ln: 1
Col: 0

```
#Name:-Hardik D Vaghasiya
#Roll No:-4044
#Class:-SYCS
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)
```



```
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 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:\ROUGH\New folder\Desktop\ds assignment\Practical-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
>>> 
```