**NAME : RAJVARDHAN REDDY**

**REG NO : 180905093**

**SEC : B – B1 , ROLL : 19**

**P1)**

class py\_solution:

def sub\_sets(self, sset):

return self.subsetsRecur([], sorted(sset))

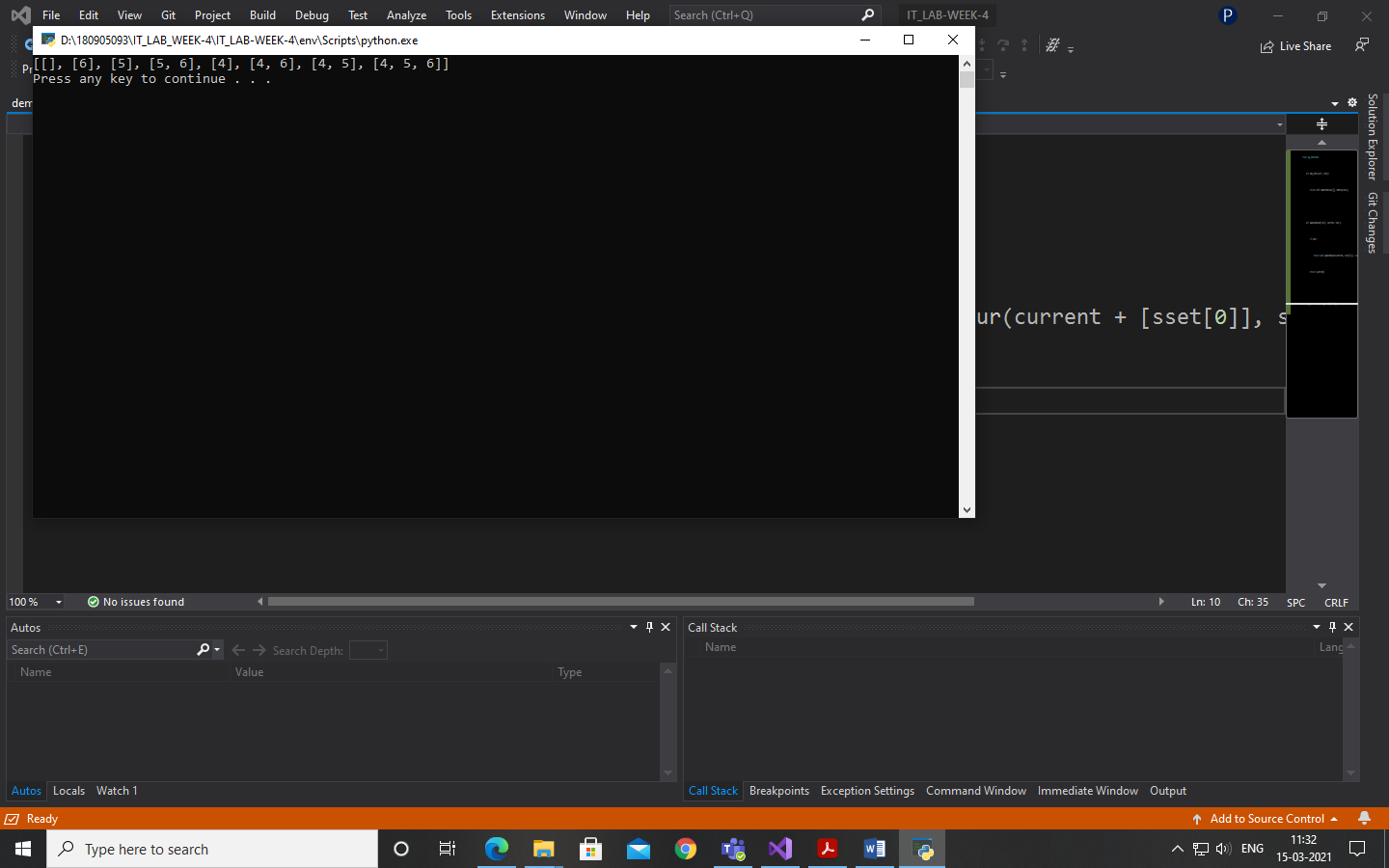
def subsetsRecur(self, current, sset):

if sset:

return self.subsetsRecur(current, sset[1:]) + self.subsetsRecur(current + [sset[0]], sset[1:])

return [current]

print(py\_solution().sub\_sets([4,5,6]))



**P2)**

class lab4p2:

def pairSum(arr, n, sum):

res = []

for i in range(0, n):

j = 0

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

if (arr[j] + arr[i]) == sum:

res.append([i+1, j+1])

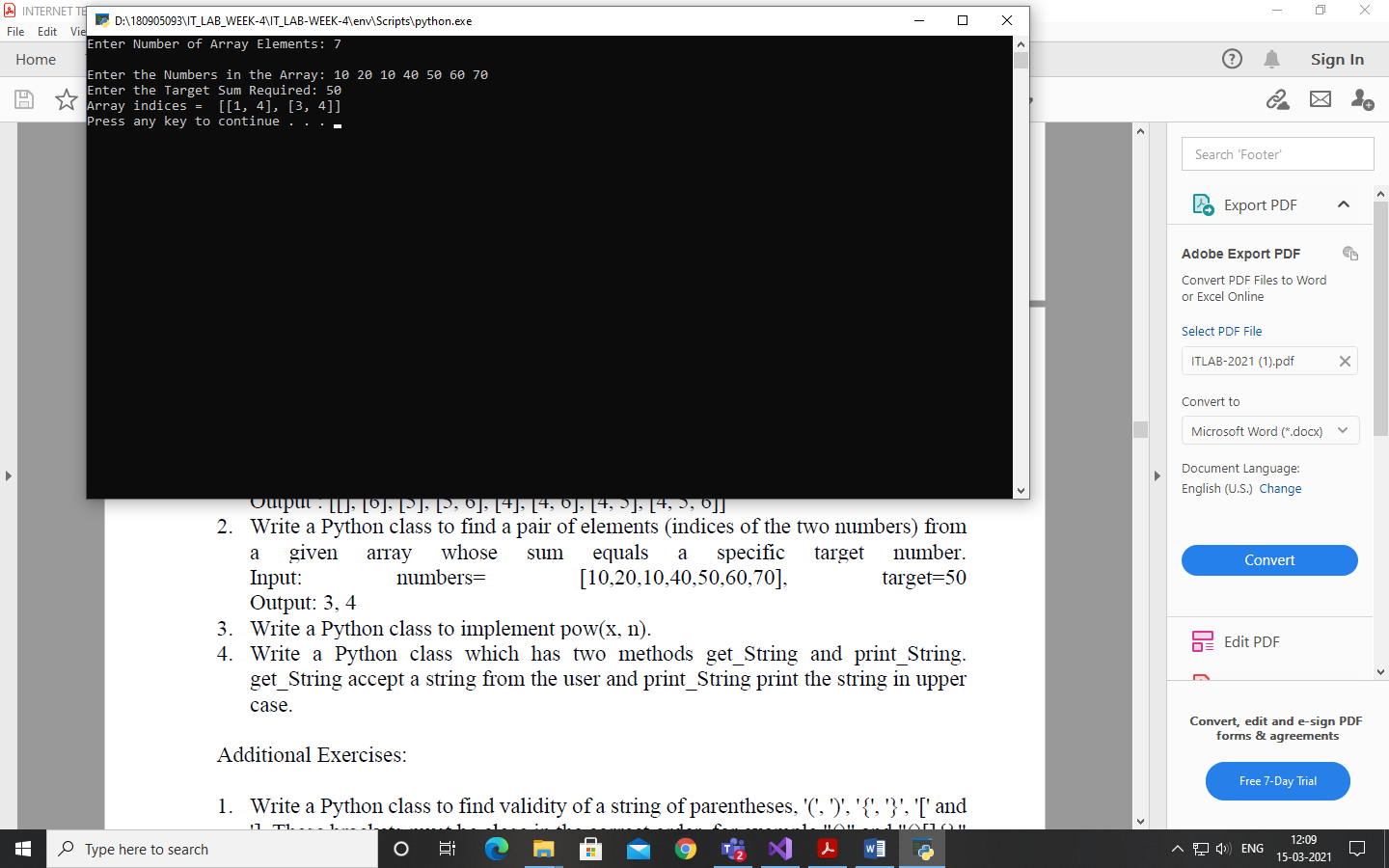
return res

n = int(input("Enter Number of Array Elements: "))

arr = list(map(int, input("\nEnter the Numbers in the Array: ").strip().split()))[:n]

sum = int(input("Enter the Target Sum Required: "))

print("Array indices = ", lab4p2.pairSum(arr, n, sum))

****

**P3)**

class py\_solution:

def pow(self, x, n):

if x==0 or x==1 or n==1:

return x

if x==-1:

if n%2 ==0:

return 1

else:

return -1

if n==0:

return 1

if n<0:

return 1/self.pow(x,-n)

val = self.pow(x,n//2)

if n%2 ==0:

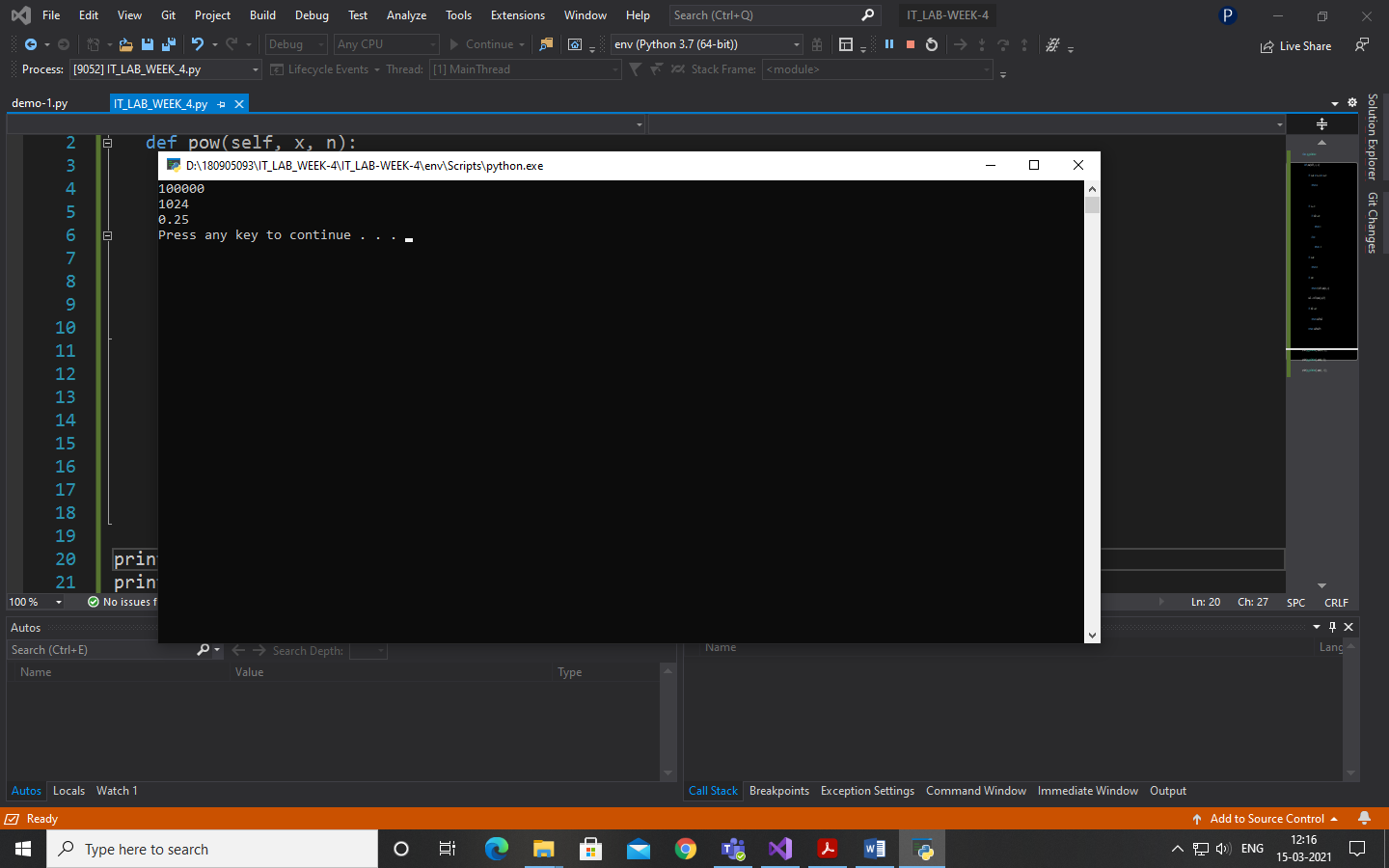
return val\*val

return val\*val\*x

print(py\_solution().pow(10, 5));

print(py\_solution().pow(4, 5));

print(py\_solution().pow(2, -2));



**P4)**

class IOString():

def \_\_init\_\_(self):

self.str1 = ""

def get\_String(self):

self.str1 = input()

def print\_String(self):

print(self.str1.upper())

str1 = IOString()

str1.get\_String()

str1.print\_String()

