DAY-3 LAB EXPERIMENTS

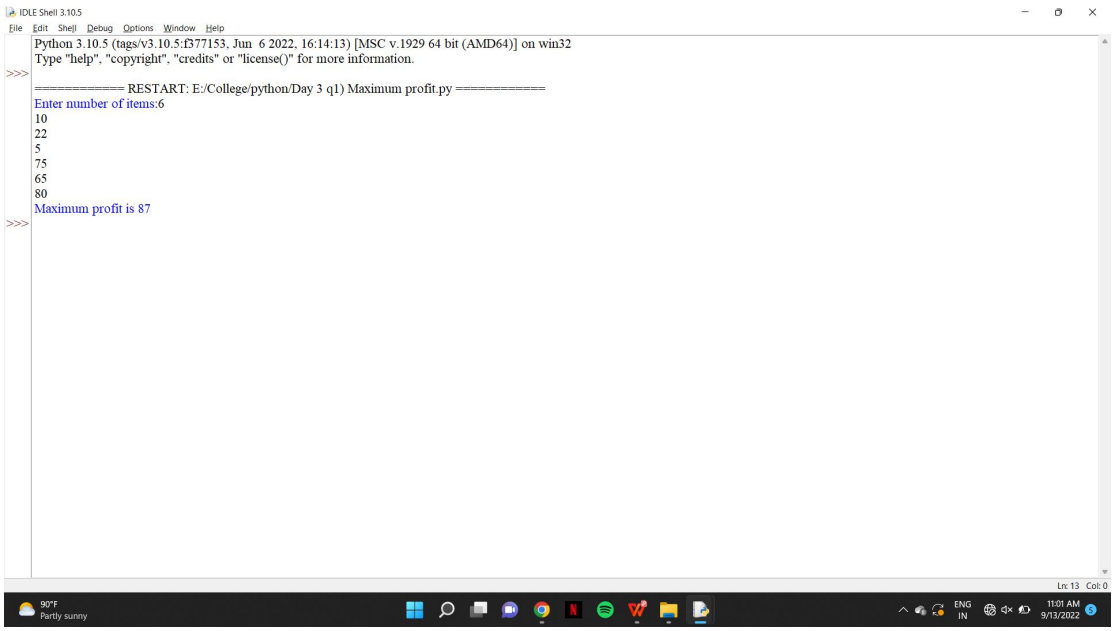
NAME: S.G.DEVSACHIN
REG.NO: 192111088
SUBJECT CODE:CSA0836
SUBJECT: PYTHON PROGRAMMING
DATE: 13/09/2022

1) PROGRAM:

```
def maxProfit(price, n):

 profit = [0]*n

 max_price = price[n-1]

 for i in range(n-2, 0, -1):

  if price[i] > max_price:
   max_price = price[i]

  profit[i] = max(profit[i+1], max_price - price[i])

 min_price = price[0]

 for i in range(1, n):

  if price[i] < min_price:
   min_price = price[i]

  profit[i] = max(profit[i-1], profit[i]+(price[i]-min_price))

 result = profit[n-1]

 return result
a=[]
n=int(input("Enter number of items:"))
```

```
for i in range(n):
    a.append(int(input()))
print ("Maximum profit is", maxProfit(a, len(a)))
```

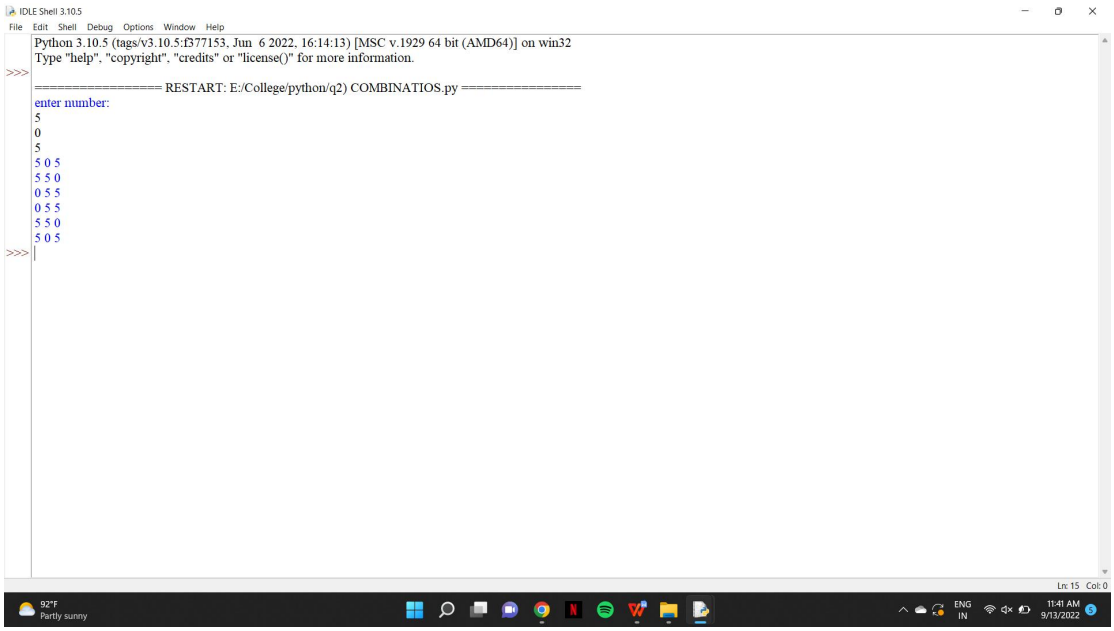
OUTPUT:



2) def comb(L):

```
    for i in range(3):
        for j in range(3):
            for k in range(3):

                # check if the indexes are not
                # same
                if (i!=j and j!=k and i!=k):
                    print(L[i], L[j], L[k])

a=[]
print("enter number:")
for i in range(3):
    b=int(input())
    a.append(b)
comb(a)
```

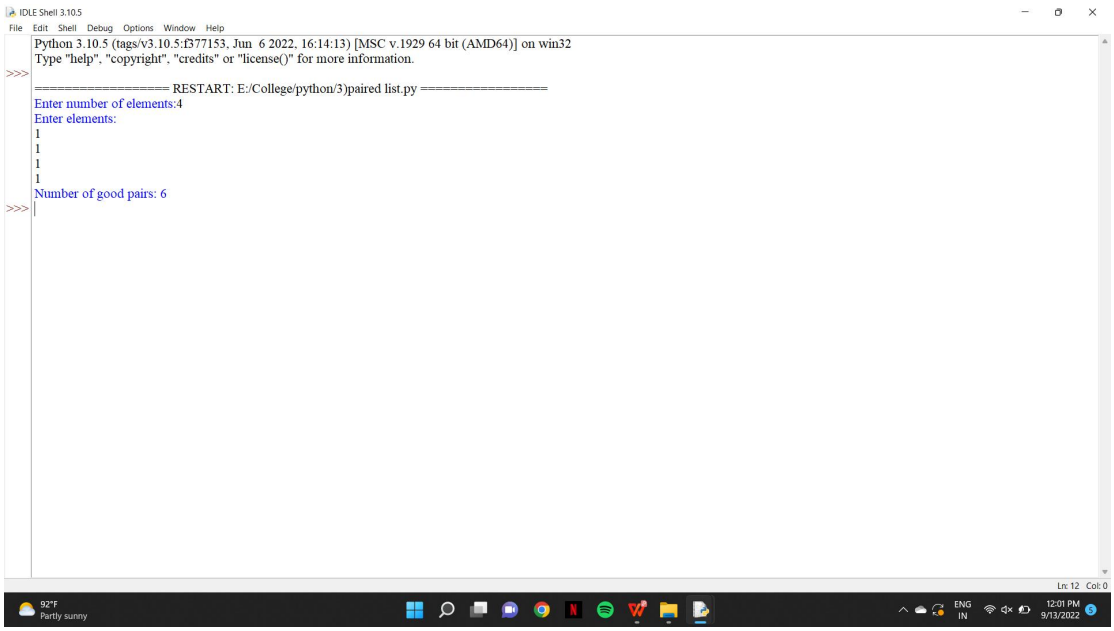
OUTPUT:



3) def solve(nums):
    count=0
    n=len(nums)
    for i in range(n):
        for j in range(i+1,n):
            if nums[i] == nums[j]:
                count+=1
    return count

a=[]
n=int(input("Enter number of elements:"))
print("Enter elements:")
for i in range(n):
        b=int(input())
        a.append(b)
print("Number of good pairs:",solve(a))

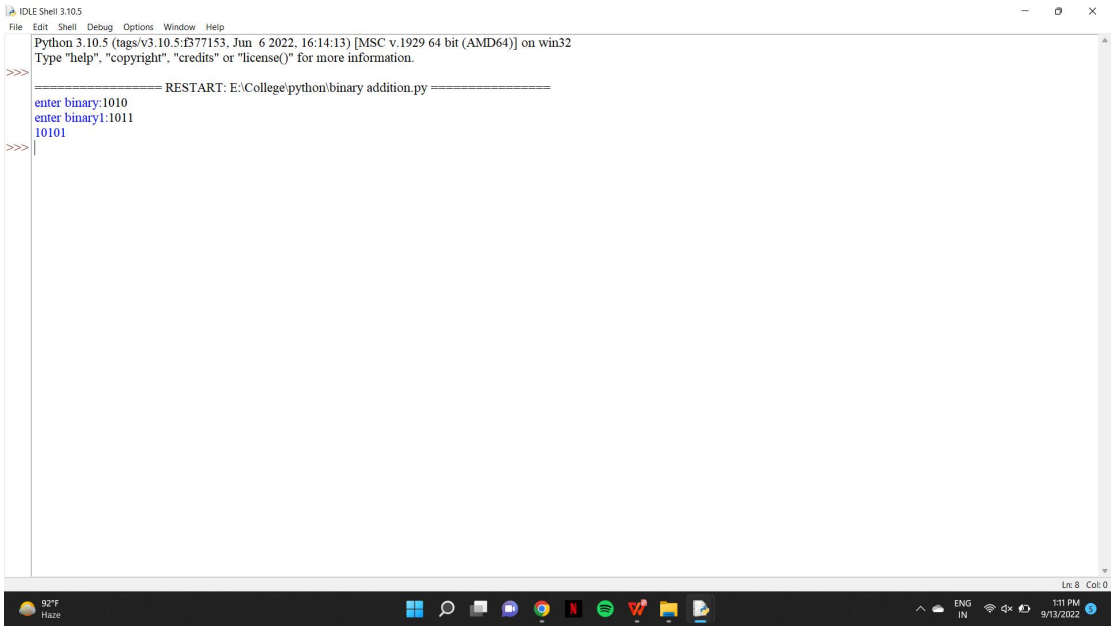
OUTPUT:



4) a = input("enter binary:")
b = input("enter binary1:")
sum = bin(int(a, 2) + int(b, 2))

# Printing result
print(sum[2:])

OUTPUT:

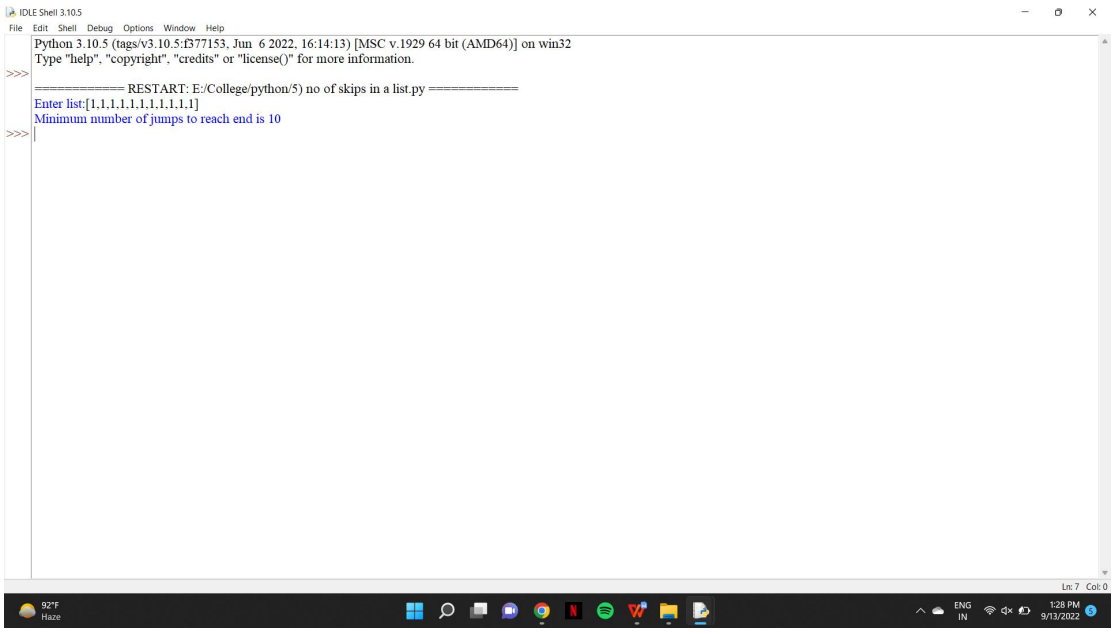


5) def minJumps(arr, l, h):

if (h == l):

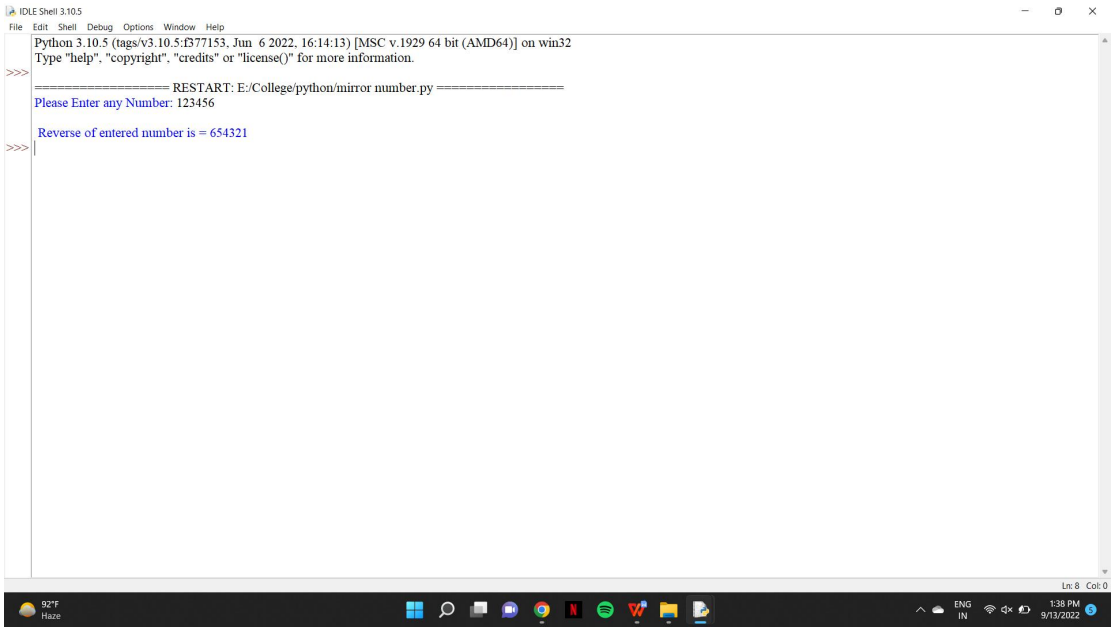
```
 return 0

if (arr[l] == 0):
 return float('inf')

min = float('inf')
for i in range(l + 1, h + 1):
 if (i < l + arr[l] + 1):
  jumps = minJumps(arr, i, h)
  if (jumps != float('inf') and
    jumps + 1 < min):
   min = jumps + 1

 return min
arr=eval(input("Enter list:"))
n=len(arr)
print('Minimum number of jumps to reach',
 'end is', minJumps(arr, 0, n-1))
```

OUTPUT:



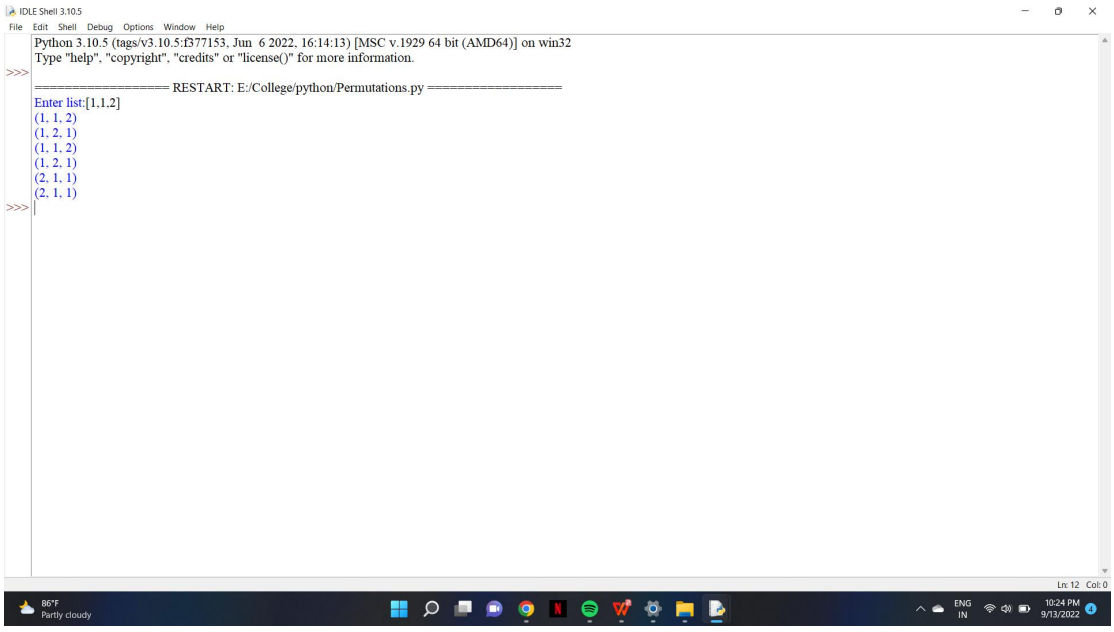

```
6) Number = int(input("Please Enter any Number: "))
Reverse = 0
while(Number > 0):
    Reminder = Number %10
    Reverse = (Reverse *10) + Reminder
    Number = Number //10
    print("\n Reverse of entered number is = Reverse)
```

OUTPUT:




```
7) from itertools import permutations
a=eval(input("Enter list:"))
per = permutations(a)
for i in list(per):
    print(i)
```

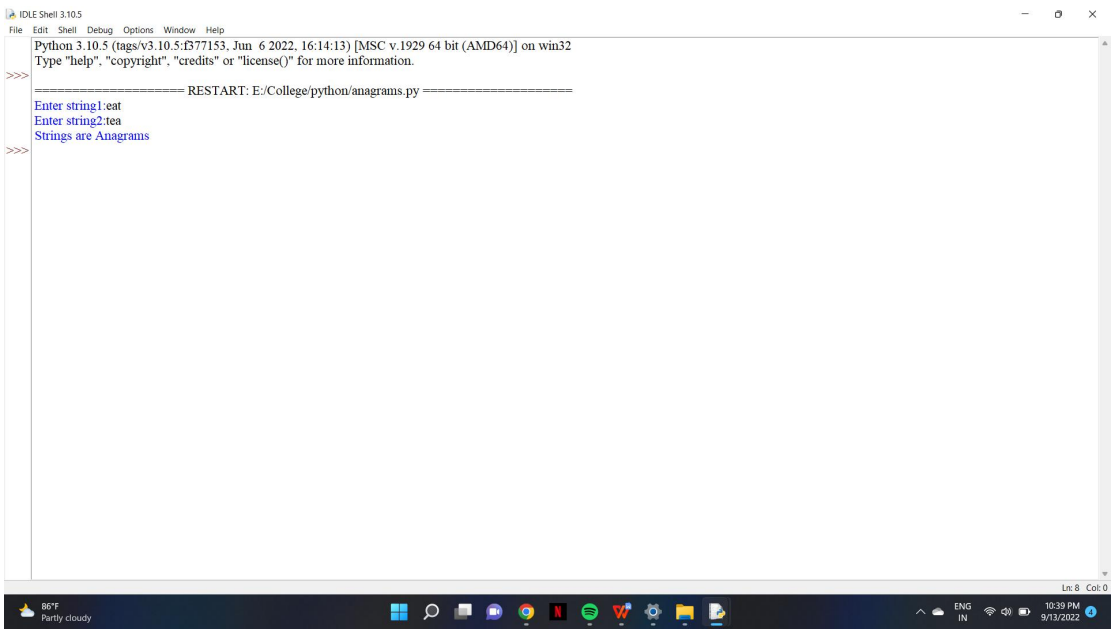
OUTPUT:




```
8) str1=input("Enter string1:")
str2=input("Enter string2:")
```

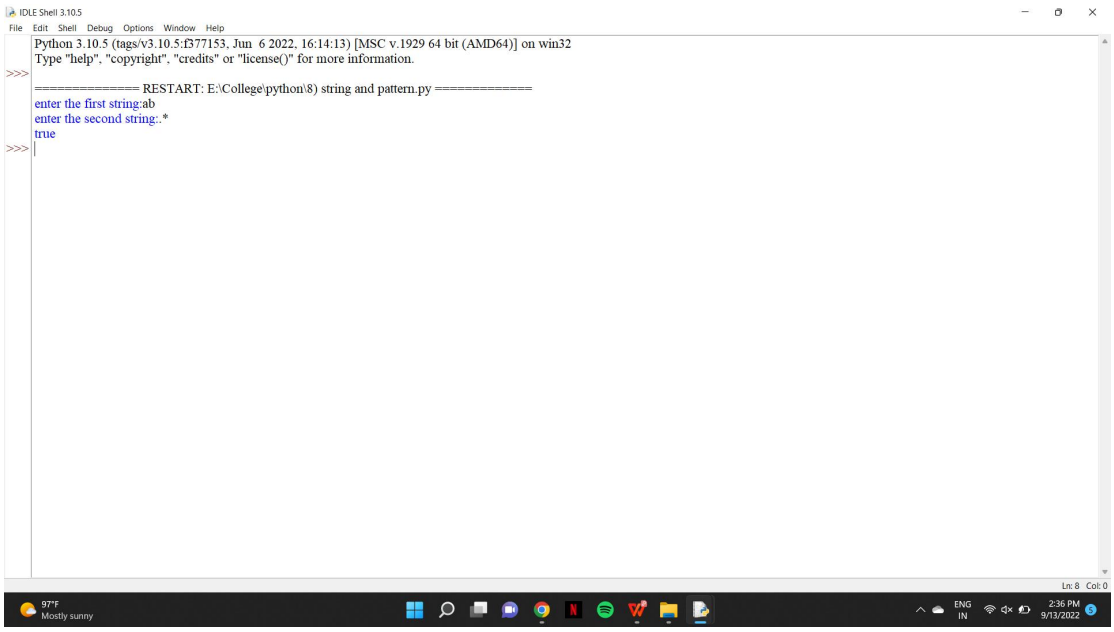
```
if len(str1)!=len(str2):
    print("Not Anagrams")
else:
    if sorted(str1)==sorted(str2):
        print("Strings are Anagrams")
    else:
        print("Not Anagrams")
```

OUTPUT:



9)
```
import re
s = input("enter the first string:")
p = input("enter the second string:")
p = r"{}".format(p)
p = re.compile(p)
if p.fullmatch(s):
    print("true")
else:
    print("false")
```

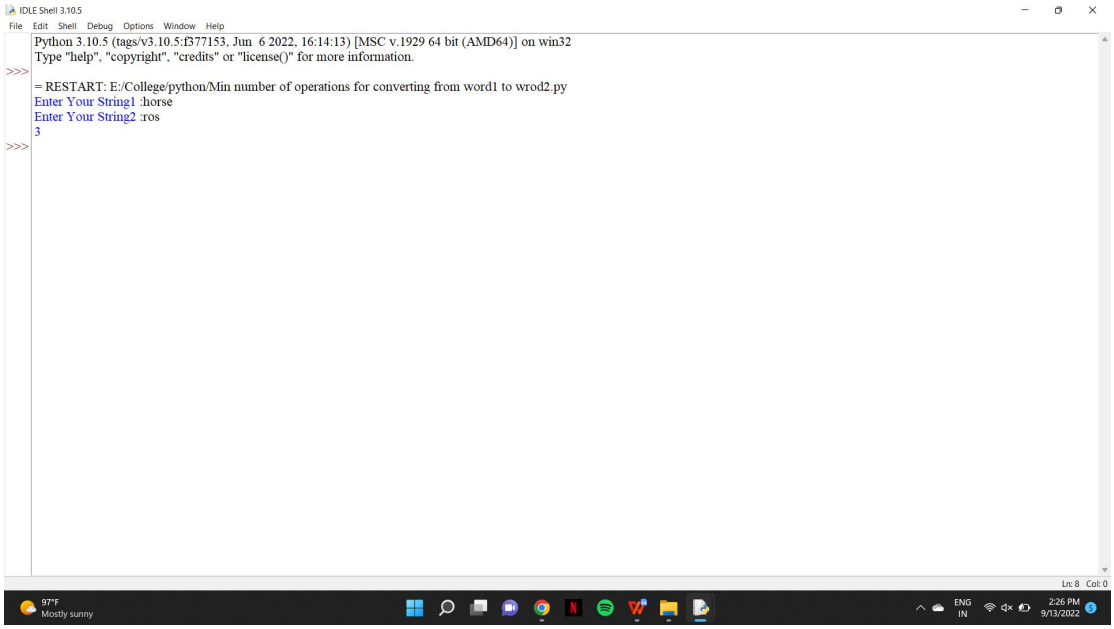
OUTPUT:



10) def editDistance(str1, str2, m, n):

```
    if m == 0:
        return n

    if n == 0:
        return m

    if str1[m-1] == str2[n-1]:
        return editDistance(str1, str2, m-1, n-1)

    return 1 + min(editDistance(str1, str2, m, n-1),
                   editDistance(str1, str2, m-1, n),
                   editDistance(str1, str2, m-1, n-1)
                   )

str1 = input("Enter Your String1 :")
str2 = input("Enter Your String2 :")
print (editDistance(str1, str2, len(str1), len(str2)))
```

OUTPUT:


IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:/College/python/Min number of operations for converting from word1 to wrod2.py
Enter Your String1 :horse
Enter Your String2 :ros
3
>>>
Ln: 8 Col: 0
97°F
Mostly sunny
ENG
IN
2:26 PM
9/13/2022