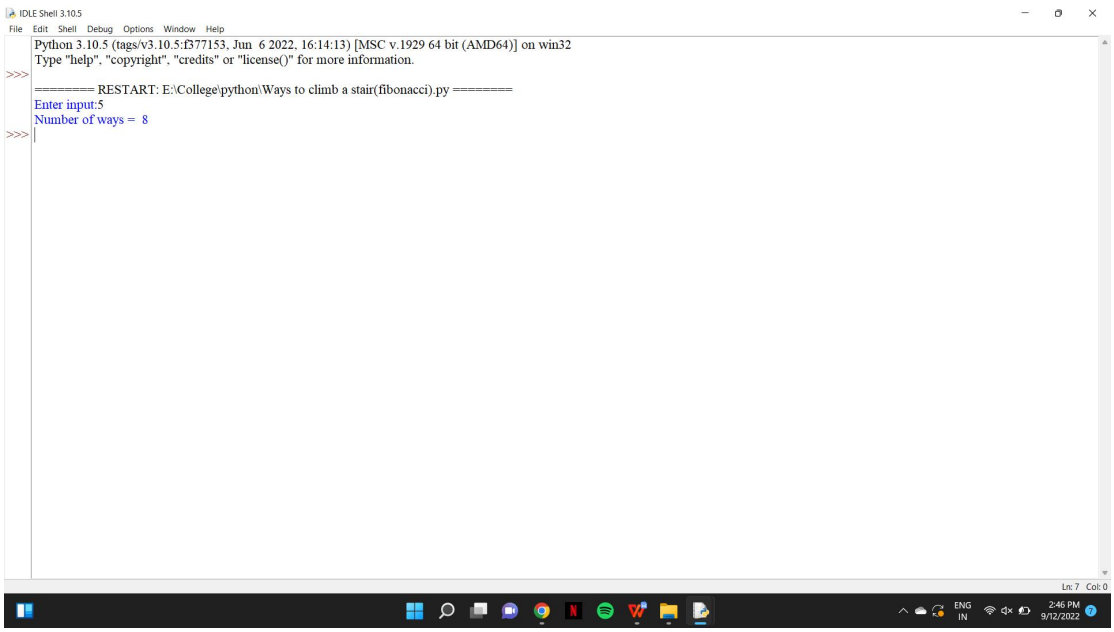
DAY 2 LAB EXPERIMENTS

NAME: S.G.DEVSACHIN
REG.NO: 192111088
SUBJECT CODE: CSA0836

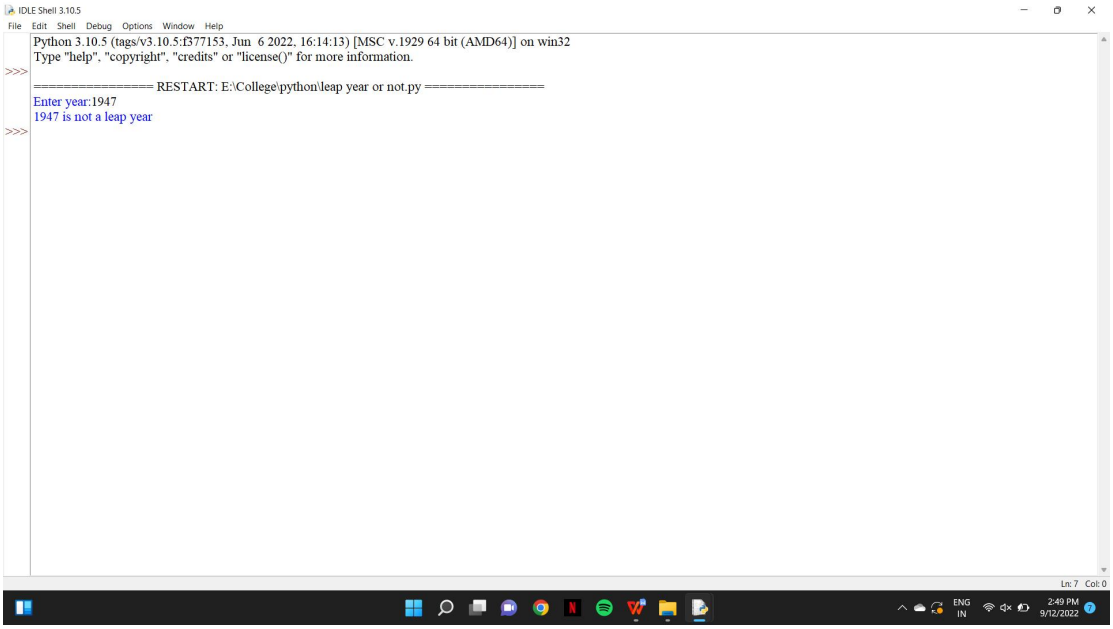
1)
```
def fib(n):
    if n <= 1:
        return n
    return fib(n-1) + fib(n-2)
def countWays(s):
    return fib(s + 1)
s = int(input("Enter input:"))
print ("Number of ways = ",countWays(s))
```

OUTPUT:



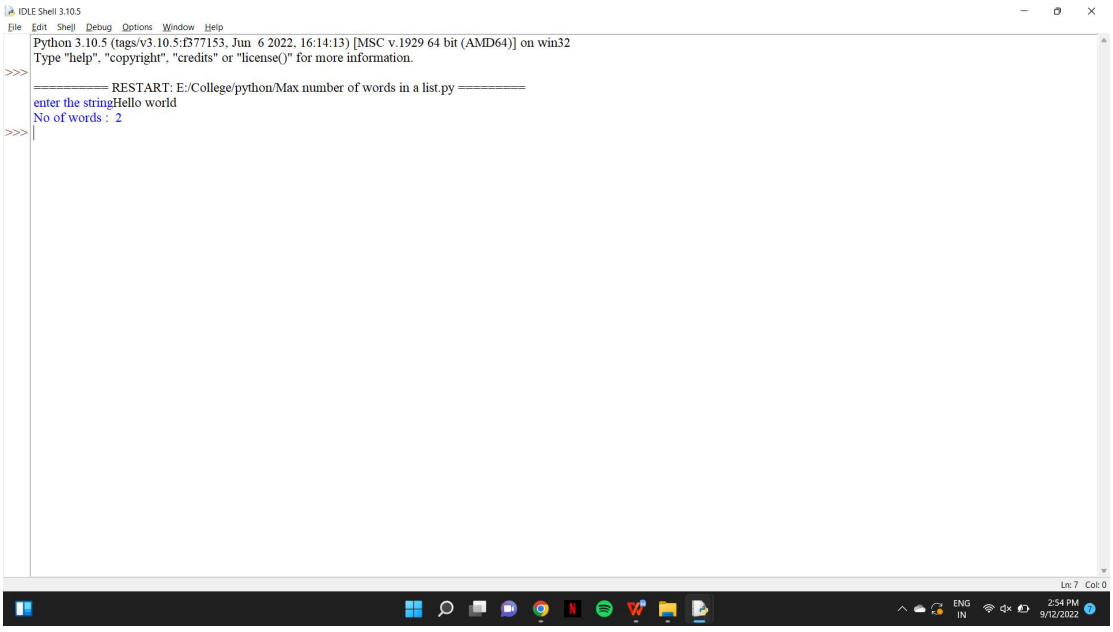
2)
```
year=int(input("Enter year:"))
if (year%400==0):
    print(year,"is a leap year")
elif (year%100!=0 and year%4==0):
    print(year,"is a leap year")
else:
    print(year,"is not a leap year")
```

OUTPUT:




```
3) def countWords(s):
        if s.strip() == "":
                return 0
        words = s.split()
        return len(words)
if _name_ == "_main_":
        s = input("enter the string")
        print("No of words : ", countWords(s))
```

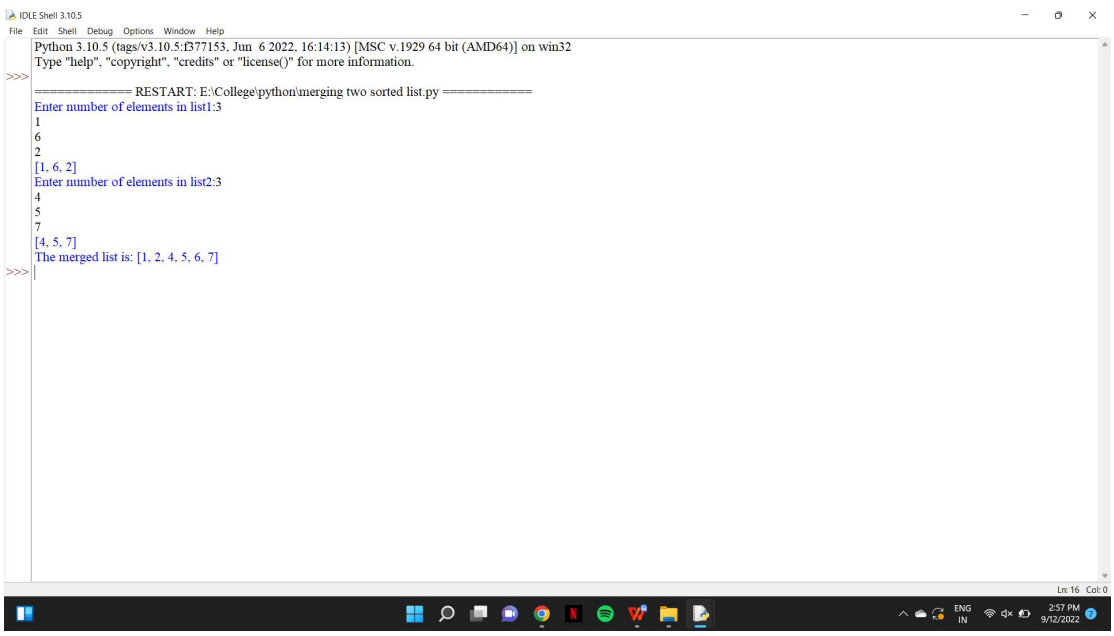
OUTPUT:



4) lst1=[]

```
lst2=[]
n1=int(input("Enter number of elements in list1:"))
for i in range(0,n1):
    a=int(input())
    lst1.append(a)
print(lst1)
n2=int(input("Enter number of elements in list2:"))
for i in range(0,n2):
    b=int(input())
    lst2.append(b)
print(lst2)
lst3=lst1+lst2
lst3.sort()
print("The merged list is:",lst3)
```

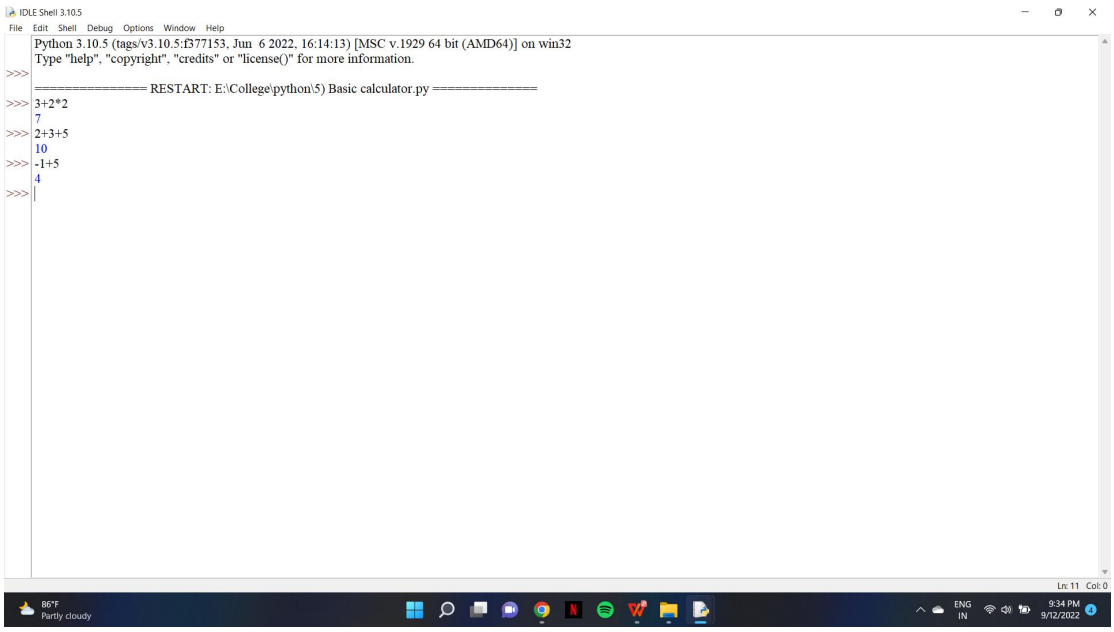
OUTPUT:



5) def calculate(self,s):

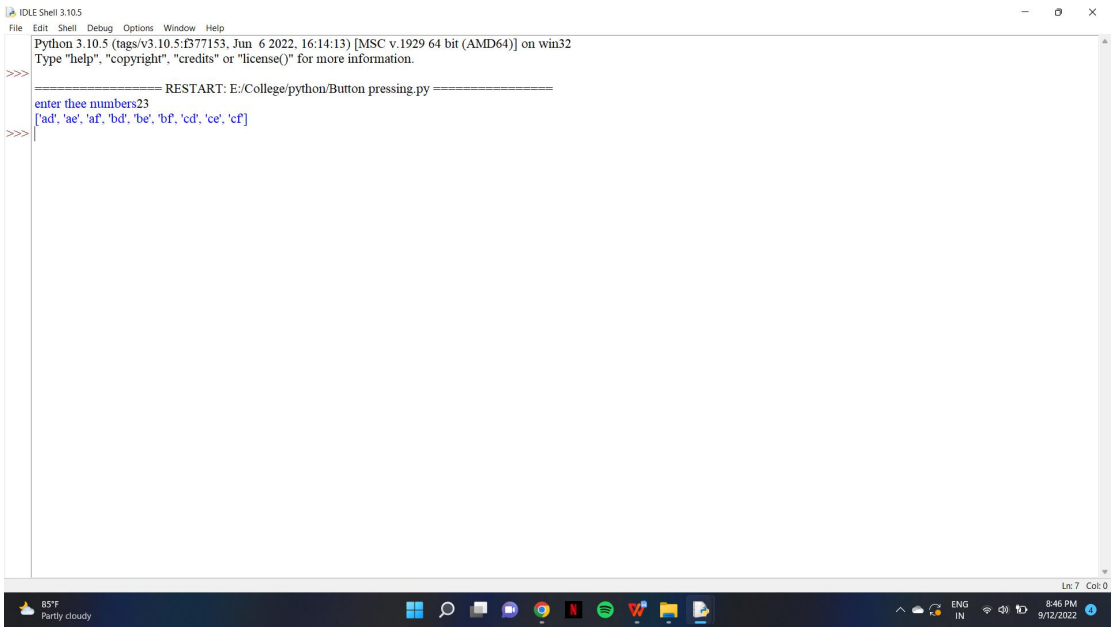
```
    """
    :type s: str
    :rtype: int
    """
```

OUTPUT:



6) class Solution(object):
  def letterCombinations(self, digits):
    if len(digits) == 0:
      return []
    characters =
{2:"abc",3:"def",4:"ghi",5:"jkl",6:"mno",7:"pqrs",8:"tuv",9:"wxyz"}
    result = []
    self.solve(digits,characters,result)
    return result
  def solve(self, digits, characters, result, current_string="",current_level
= 0):
    if current_level == len(digits):
      result.append(current_string)
      return
    for i in characters[int(digits[current_level])]:
      self.solve(digits,characters,result,current_string+i,current_level+1)
ob1 = Solution()
m=(input("enter thee numbers"))
print(ob1.letterCombinations(m))

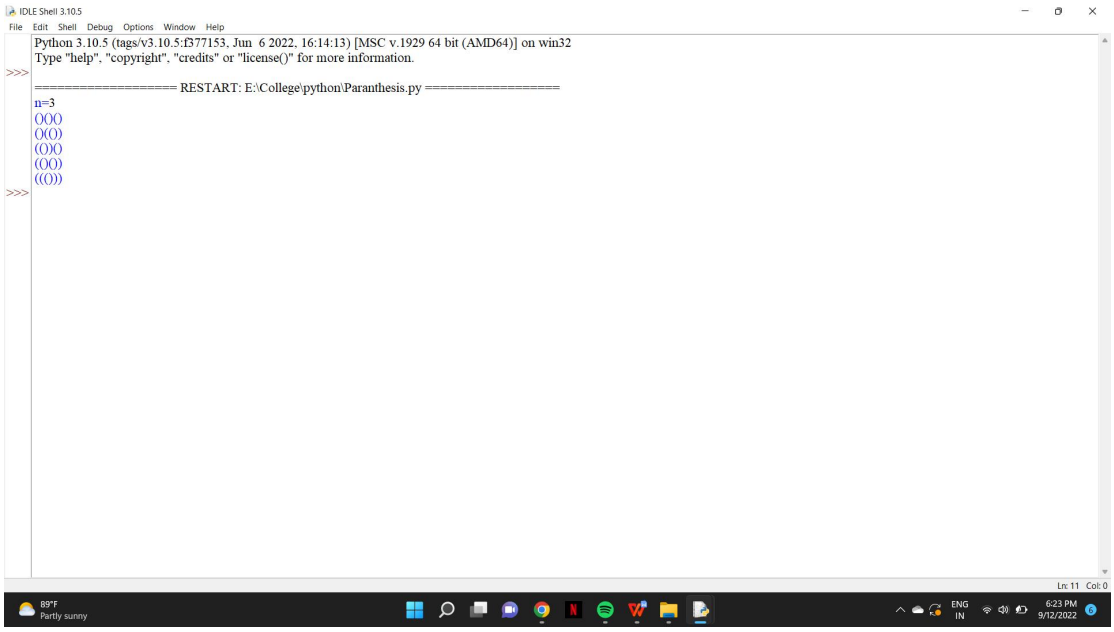
OUTPUT:



7)
```
def printParenthesis(str, n):
    if(n > 0):
        _printParenthesis(str, 0,
                          n, 0, 0)
    return


def _printParenthesis(str, pos, n,
                      open, close):

    if(close == n):
        for i in str:
            print(i, end="")
        print()
        return
    else:
        if(open > close):
            str[pos] = ')'
            _printParenthesis(str, pos + 1, n,
                              open, close + 1)
        if(open < n):
            str[pos] = '('
            _printParenthesis(str, pos + 1, n,
                              open + 1, close)
```

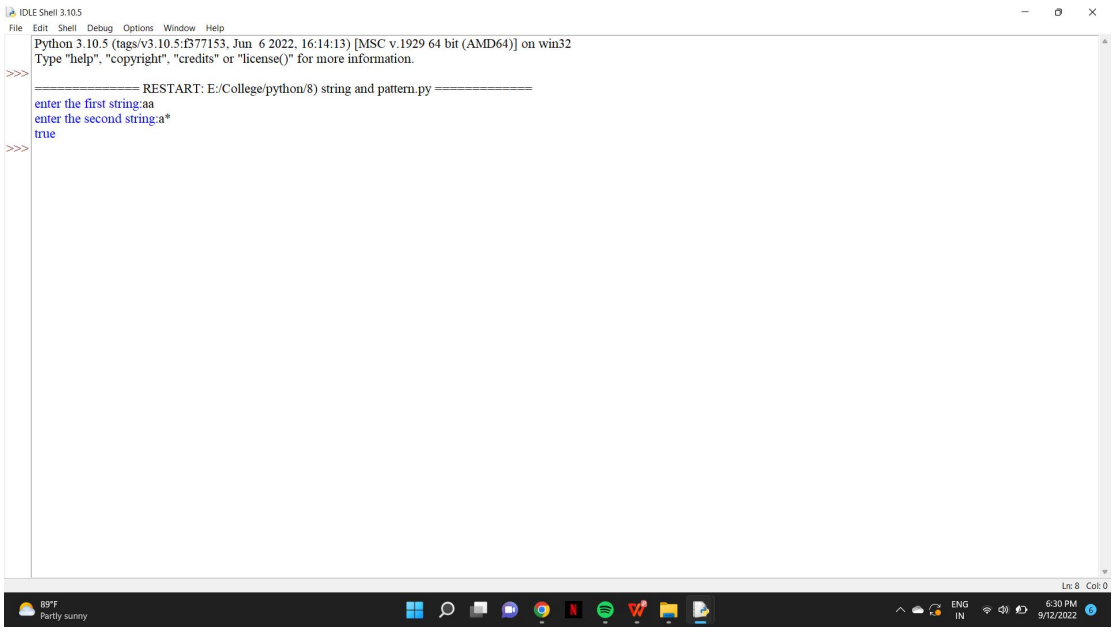
```
n = int(input("n="))
str = [" "] * 2 * n
printParenthesis(str , n)
```

OUTPUT:



8)
```
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")
```

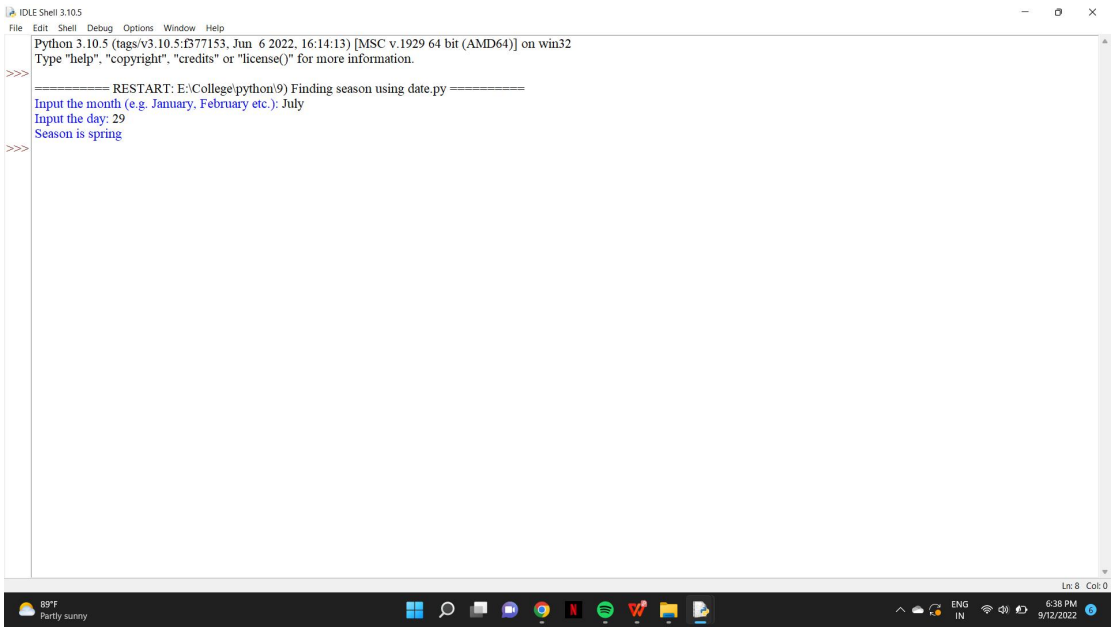
OUTPUT:



9) month = input("Input the month (e.g. January, February etc.): ")
day = int(input("Input the day: "))

```
if month in ('January', 'February', 'March'):
        season = 'winter'
elif month in ('April', 'May', 'June'):
        season = 'spring'
elif month in ('July', 'August', 'September'):
        season = 'summer'
else:
        season = 'autumn'

if (month == 'March') and (day > 19):
        season = 'spring'
elif (month == 'June') and (day > 20):
        season = 'summer'
elif (month == 'September') and (day > 21):
        season = 'autumn'
elif (month == 'December') and (day > 20):
        season = 'winter'

print("Season is",season)
```

OUTPUT:



10) # Python program for the above approach
from collections import Counter

# Function to remove common
# words from two strings
def removeCommonWords(sent1, sent2):

 # Store the words present
 # in both the sentences
 sentence1 = list(sent1.split())
 sentence2 = list(sent2.split())

 # Calculate frequency of words
 # using Counter() function frequency1 = Counter(sentence1)
 frequency2 = Counter(sentence2)

 word = 0

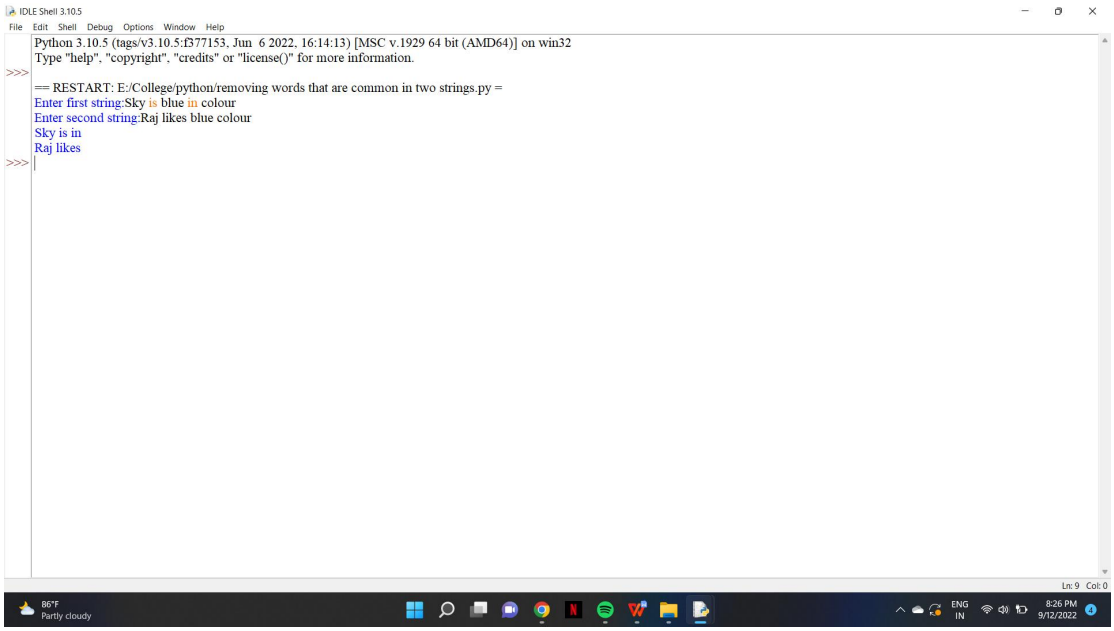
 # Iterate the list consisting
 # of words in the first sentence
for i in range(len(sentence1)):

  # If word is present # in both the strings
  if sentence1[word] in frequency2.keys():

   # Remove the word
   sentence1.pop(word)

```
  # Decrease the frequency of the word
  word = word-1
 word += 1

word = 0

# Iterate the list consisting of
# words in the second sentence
for i in range(len(sentence2)):

 # If word is present
 # in both the strings
 if sentence2[word] in frequency1.keys():

  # Remove the word
  sentence2.pop(word)

  # Decrease the removed word
  word = word-1

 word += 1

# Print the remaining
# words in the two sentences
print(*sentence1)
print(*sentence2)


# Driver Code

sentence1 = "sky is blue in color"
sentence2 = "raj likes sky blue color"

removeCommonWords(sentence1, sentence2)
```

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/removing words that are common in two strings.py =
Enter first string:Sky is blue in colour
Enter second string:Raj likes blue colour
Sky is in
Raj likes
>>>
Ln: 9 Col: 0
86°F
Partly cloudy
ENG
IN
8:26 PM
9/12/2022