

Exp-2.8

Title:

Return all strings that are substrings of another string in the list.

Aim:

To design and implement a Python program to find all strings in a list that are substrings of another string in the same list.

Procedure:

1. Read the input list of strings words.
2. Initialize an empty result list.
3. For each string in words, check if it is a substring of any other string in words (excluding itself).
4. If yes, add it to the result list.
5. Return or print the result list.

Algorithm:

1. Start
2. For each word in the list:
 - For every other word, check if the first word is contained in the second.
 - If yes, add the word to the result list and break inner loop.
3. Return the result list.
4. Stop.

Input:

4

mass as hero superhero

3

leetcode et code

3

blue green bu

Output:

['as', 'hero']

['et', 'code']

[]

Program:

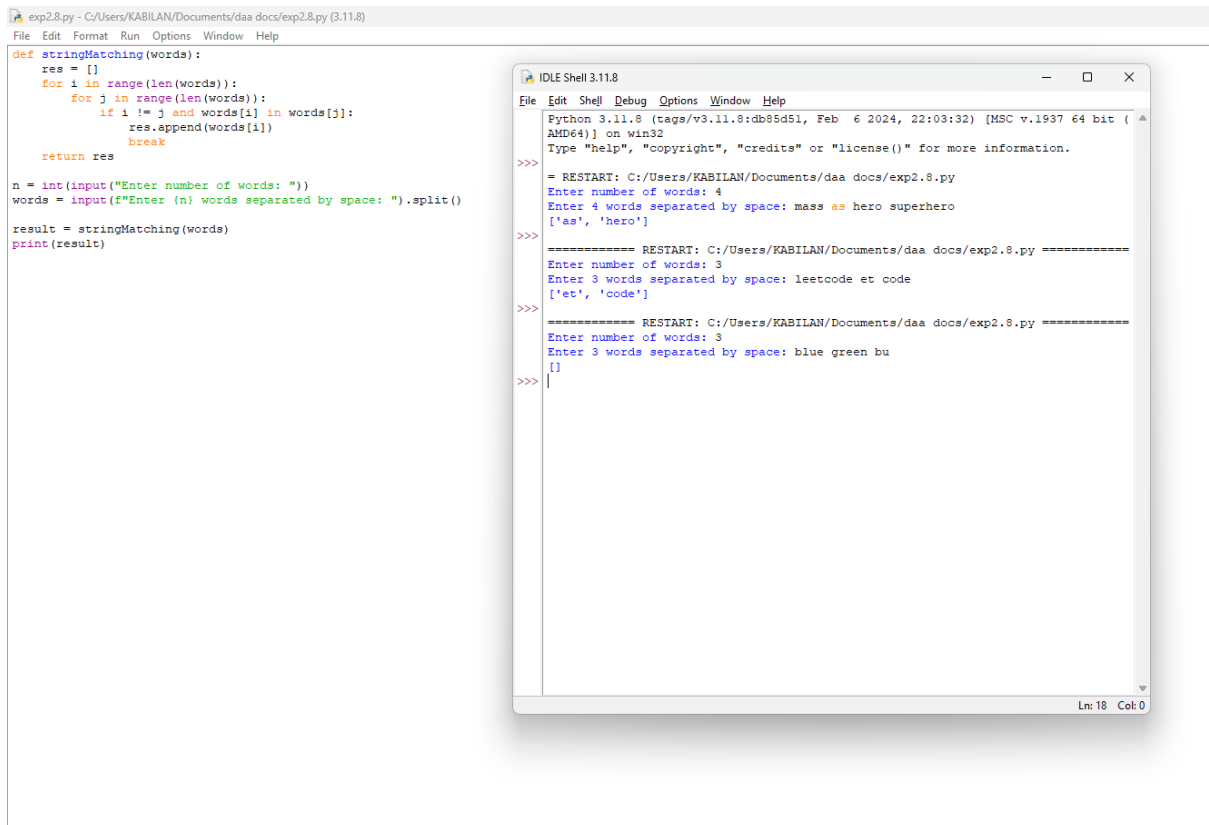
```
def stringMatching(words):  
    res = []  
    for i in range(len(words)):  
        for j in range(len(words)):  
            if i != j and words[i] in words[j]:  
                res.append(words[i])  
                break  
    return res  
  
n = int(input("Enter number of words: "))  
words = input(f"Enter {n} words separated by space: ").split()  
  
result = stringMatching(words)  
print(result)
```

Performance Analysis:

Time Complexity: $O(n^2 * k)$

Space Complexity: $O(n)$

Program Output:



The image shows a Python IDE with two windows. The left window displays the source code for a program named 'exp2.8.py'. The code defines a function 'stringMatching(words)' that finds all substrings of a given list of words. It then prompts the user for the number of words and the words themselves, and prints the result of the function.

```
exp2.8.py - C:/Users/KABILAN/Documents/daa docs/exp2.8.py (3.11.8)
File Edit Format Run Options Window Help
def stringMatching(words):
    res = []
    for i in range(len(words)):
        for j in range(len(words)):
            if i != j and words[i] in words[j]:
                res.append(words[i])
                break
    return res

n = int(input("Enter number of words: "))
words = input(f"Enter {n} words separated by space: ").split()
result = stringMatching(words)
print(result)
```

The right window shows the execution of the program. It displays the prompts and user input, followed by the output of the function.

```
IDLE Shell 3.11.8
File Edit Shell Debug Options Window Help
Python 3.11.8 (tags/v3.11.8:db85d51, Feb 6 2024, 22:03:32) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/KABILAN/Documents/daa docs/exp2.8.py
Enter number of words: 4
Enter 4 words separated by space: mass as hero superhero
['as', 'hero']

===== RESTART: C:/Users/KABILAN/Documents/daa docs/exp2.8.py =====
Enter number of words: 3
Enter 3 words separated by space: leetcode et code
['et', 'code']

===== RESTART: C:/Users/KABILAN/Documents/daa docs/exp2.8.py =====
Enter number of words: 3
Enter 3 words separated by space: blue green bu
[]
>>>
```

Result:

Thus the given program Substring Words Finder is executed and got output successfully.