Started on	Friday, 5 April 2024, 12:53 PM
State	Finished
Completed on	Thursday, 11 April 2024, 11:18 AM
Time taken	5 days 22 hours
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	AVULA SNEYA DRITI 2022-CSD-A

Question **1**

Correct

Mark 1.00 out of 1.00

Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

Sample Input 1

thistest123string

123

Sample Output 1

8

Answer: (penalty regime: 0 %)

```
string1 = input("")
string2 = input("")

found_index = -1

for i in range(len(string1)):
    if string1[i:i+len(string2)] == string2:
        found_index = i
        break

print(found_index)
```

	Input	Expected	Got	
✓	thistest123string 123	8	8	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Consider the below words as key words and check the given input is key word or not.

keywords: {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Input format:

Take string as an input from stdin.

Output format:

Print the word is key word or not.

Example Input:

break

Output:

break is a keyword

Example Input:

IF

Output:

IF is not a keyword

For example:

Input	**Result**
`break`	`break is a keyword`
`IF`	`IF is not a keyword`

Answer: (penalty regime: 0 %)

```
keywords = ["break", "case", "continue", "default", "defer", "else", "for", "f

input_word = input("")  # Convert input to lowercase for case-insensitive comp

if input_word in keywords:
    print(input_word + " is a keyword")
else:
    print(input_word + " is not a keyword")
```

	Input	**Expected**	**Got**	
✓	`break`	`break is a keyword`	`break is a keyword`	✓

	Input	Expected	Got	
✓	`IF`	`IF is not a keyword`	`IF is not a keyword`	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Consider the below words as key words and check the given input is key word or not.

keywords: {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Input format:

Take string as an input from stdin.

Output format:

Print the word is key word or not.

Example Input:

break

Output:

break is a keyword

Example Input:

IF

Output:

IF is not a keyword

For example:

Input	Result
break	break is a keyword
IF	IF is not a keyword

Answer: (penalty regime: 0 %)

```
keywords = ["break", "case", "continue", "default", "defer", "else", "for", "f

input_word = input("")  # Convert input to lowercase for case-insensitive comp

if input_word in keywords:
    print(input_word + " is a keyword")
else:
    print(input_word + " is not a keyword")
```

	Input	Expected	Got	
✓	break	break is a keyword	break is a keyword	✓

	Input	Expected	Got	
✓	`IF`	`IF is not a keyword`	`IF is not a keyword`	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Verify the given number is cyclic or not.

Input Format

Num1

Num2

Constraints

1<=range<=9999999999

Sample Input 1

12345

45123

Sample Output 1

Yes

Sample Input 2

12345

54123

Sample Output 2

No

Answer: (penalty regime: 0 %)

```python
num1_1 = int(input(""))
num2_1 = int(input(""))
num1_str = str(num1_1)
num2_str = str(num2_1)

if len(num1_str) != len(num2_str):
    print("No")
else:
    double_num1 = num1_str + num1_str
    if num2_str in double_num1:
        print("Yes")
    else:
        print("No")

```

	Input	Expected	Got	
✓	12345 45123	Yes	Yes	✓
✓	12345 54123	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Balanced strings are those that have an equal quantity of 'L' and 'R' characters.

Given a balanced string s, split it in the maximum amount of balanced strings.

Return the maximum amount of split balanced strings.

Example 1:

Input:

RLRRLLRLRL

Output:

4

Explanation: s can be split into "RL", "RRLL", "RL", "RL", each substring contains same number of 'L' and 'R'.

Example 2:

Input:

RLLLLRRRLR

Output:

3

Explanation: s can be split into "RL", "LLLRRR", "LR", each substring contains same number of 'L' and 'R'.

Example 3:

Input:

LLLLRRRR

Output:

1

Explanation: s can be split into "LLLLRRRR".

Constraints:

1 <= s.length <= 1000

s[i] is either 'L' or 'R'.

s is a balanced string.

Answer: (penalty regime: 0 %)

```
s1 = input("")
count1 = 0
balance1 = 0

for char in s1:
    if char == 'L':
        balance1 += 1
    else:
        balance1 -= 1

    if balance1 == 0:
        count1 += 1

print(count1)
```

	Input	Expected	Got	
✓	RLRRLLRLRL	4	4	✓
✓	RLLLLRRRLR	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.