Started on	Tuesday, 7 May 2024, 6:19 PM
State	Finished
Completed on	Tuesday, 7 May 2024, 6:24 PM
Time taken	5 mins 16 secs
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	DWIJESH SREERAM S 2022-CSD-A

Question **1**

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', 'func', 'goto', 'if', '
input= input()
if input in keywords:
    print(input, "is a keyword")
else:
    print(input, "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 **2**

Correct

Mark 1.00 out of 1.00

Write a Python program to get one string and reverses a string. The input string is given as an array of characters `char[]`.

You may assume all the characters consist of printable ascii characters.

Example 1:

```
Input:
hello
Output:
olleh
```

Example 2:

```
Input:
Hannah
Output:
hannaH
```

Answer: (penalty regime: 0 %)

```
def reverse_string(s):
    left, right = 0, len(s) - 1
    while left < right:
        s[left], s[right] = s[right], s[left]
        left += 1
        right -= 1

# Example usage:
input_string = list(input())
reverse_string(input_string)
print(''.join(input_string))

```

	Input	Expected	Got	
✔	hello	olleh	olleh	✔
✔	Hannah	hannaH	hannaH	✔

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', 'func', 'goto', 'if', '
input= input()
if input in keywords:
    print(input, "is a keyword")
else:
    print(input, "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 %)

```
def is_cyclic(num1, num2):

    double_num1 = num1 + num1

    return num2 in double_num1


num1 = input()
num2 = input()


if is_cyclic(num1, num2):
    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

Program :

Write a function to check whether two given strings are anagram of each other or not. An anagram of a string is another string that contains the same characters, only the order of characters can be different. For example, “abcd” and “dabc” are an anagram of each other.

Given two strings s1 and s2, check if both the strings are anagrams of each other.

If both strings are anagrams print as "true", otherwise display as "false"
Examples:

```
Input : s1 = "listen"
```

```
        s2 = "silent"
```

```
Output : true
```

For example:

Input	Result
dad bad	false

Answer: (penalty regime: 0 %)

```
def are_anagrams(s1, s2):
    # Remove spaces and convert strings to lowercase
    s1 = s1.replace(" ", "").lower()
    s2 = s2.replace(" ", "").lower()

    # Sort the characters in both strings
    sorted_s1 = sorted(s1)
    sorted_s2 = sorted(s2)

    # Compare the sorted strings
    return sorted_s1 == sorted_s2

# Example usage:
s1 = input()
s2 = input()

if are_anagrams(s1, s2):
    print("true")
else:
    print("false")

```

	Input	Expected	Got	
✔	`listen` `silent`	`true`	`true`	✔
✔	`dad` `bad`	`false`	`false`	✔
✔	`triangle` `integral`	`true`	`true`	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ Week-06_MCQ

Jump to...

WEEK-06-Extra ►