Solutions,28jan2022

1.Write a Python program to check the nth-1 string is a proper substring of nth string in a given list of strings.

Solution:

```
>>> def test(str1):
>>>     return str1[len(str1)-2] in str1[len(str1)-1] and str1[len(str1)-2] != str1[len(str1)-1]
>>>
>>> str11 = ["a","abb","sfs", "oo", "de", "sfde"]
>>> print("Original list:")
>>> print(str11)
>>> print("Check the nth-1 string is a proper substring of nth string of the said list of strings:")
>>> print(test(str11))
>>> str11 = ["a","abb","sfs", "oo", "ee", "sfde"]
>>> print("\nOriginal list:")
>>> print(str11)
>>> print("Check the nth-1 string is a proper substring of nth string of the said list of strings:")
>>> print(test(str11))
>>> str11 = ["a","abb","sad", "ooaa" "esdfe", "sfsdfde", "sfsd", "sfsdf", "qwrew"]
>>> print("\nOriginal list:")
>>> print(str11)
>>> print("Check the nth-1 string is a proper substring of nth string of the said list of strings:")
>>> print(test(str11))
>>> str11 = ["a","abb","sad", "ooaa" "esdfe", "sfsdfde", "sfsd", "sfsdf", "qwsfsdfrew"]
>>> print("\nOriginal list:")
>>> print(str11)
>>> print("Check the nth-1 string is a proper substring of nth string of the said list of strings:")
>>> print(test(str11))
```

2.We are making n stone piles! The first pile has n stones. If n is even, then all piles have an even number of stones. If n is odd, all piles have an odd number of stones. Each pile must more stones than the previous pile but as few as possible. Write a Python program to find the number of stones in each pile.

Solution:

```
>>> def test(n):
>>>     return [n + 2 * i for i in range(n)]
>>> n = 2
>>> print("Number of piles:",n)
>>> print("Number of stones in each pile:")
>>> print(test(n))
>>> n = 10
>>> print("\nNumber of piles:",n)
>>> print("Number of stones in each pile:")
>>> print(test(n))
>>> n = 3
>>> print("\nNumber of piles:",n)
>>> print("Number of stones in each pile:")
>>> print(test(n))
>>> n = 17
>>> print("\nNumber of piles:",n)
>>> print("Number of stones in each pile:")
>>> print(test(n))
```

3.Write a Python program to check a given list of integers where the sum of the first i integers is i.

Solution:

```
>>> def test(nums):
>>>     return all([sum(nums[:i]) == i for i in range(len(nums))])
>>> nums = [0,1,2,3,4,5]
>>> print("Original list:")
>>> print(nums)
>>> print("Check the said list, where the sum of the first i integers is i:")
>>> print(test(nums))
>>> nums = [1,1,1,1,1,1]
>>> print("\nOriginal list:")
>>> print(nums)
>>> print("Check the said list, where the sum of the first i integers is i:")
>>> print(test(nums))
>>> nums = [2,2,2,2,2]
>>> print("\nOriginal list:")
>>> print(nums)
>>> print("Check the said list, where the sum of the first i integers is i:")
>>> print(test(nums))
```

4.Write a Python program to split a string of words separated by commas and spaces into two lists, words and separators.

Solution:

```
>>> def test(string):
>>>     import re
>>>     merged = re.split(r"([ ,]+)", string)
>>>     return [merged[::2], merged[1::2]]
>>> s = "The dance, held in the school gym, ended at midnight."
>>> print("\nOriginal string:",s)
>>> print("Split the said string into 2 lists: words and separators:")
>>> print(test(s))
>>> s = "The colors in my studyroom are blue, green, and yellow."
>>> print("\nOriginal string:",s)
>>> print("Split the said string into 2 lists: words and separators:")
>>> print(test(s))
```

5.Write a Python program to find list integers containing exactly four distinct values, such that no integer repeats twice consecutively among the first twenty entries.

Note: The list needs to have length greater than ten

Solution:

```
>>> def test(nums):
>>>     return all([nums[i] != nums[i + 1] for i in range(len(nums)-1)]) and len(set(nums)) == 4
>>> nums = [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
>>> print("Original list:")
>>> print(nums)
>>> print("Check said list of integers containing exactly four distinct values, such that no integer repeats  twice consecutively:")
>>> print(test(nums))
>>> nums = [1, 2, 3, 3, 1, 2, 3, 3, 1, 2, 3, 3, 1, 2, 3, 3]
>>> print("\nOriginal list:")
>>> print(nums)
>>> print("Check said list of integers containing exactly four distinct values, such that no integer repeats  twice consecutively:")
>>> print(test(nums))
>>> nums = [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
>>> print("\nOriginal list:")
>>> print(nums)
>>> print("Check said list of integers containing exactly four distinct values, such that no integer repeats  twice consecutively:")
```

```
>>> print(test(nums))
```

6.Given a string consisting of whitespace and groups of matched parentheses, write a Python program to split it into groups of perfectly matched parentheses without any whitespace.

Solution:

```
>>> def test(combined):
>>>    ls = []
>>>    s2 = ""
>>>    for s in combined.replace(' ', ''):
>>>        s2 += s
>>>        if s2.count("(") == s2.count(")"):
>>>            ls.append(s2)
>>>            s2 = ""
>>>    return ls
>>> combined = '( ()) ((()()())) (()) ()'
>>> print("Parentheses string:")
>>> print(combined)
>>> print("Separate parentheses groups of the said string:")
>>> print(test(combined))
>>> combined = '() (( ( )() (  )) ) ( ())'
>>> print("\nParentheses string:")
>>> print(combined)
>>> print("Separate parentheses groups of the said string:")
>>> print(test(combined))
```

7.Write a Python program find the longest string of a given list of strings.

Solution:

```
>>> def test(words):
>>>     return  max(words, key=len)
>>> strs =  ['cat', 'car', 'fear', 'center']
>>> print("Original strings:")
```

```
>>> print(strs)
>>> print("Longest string of the said list of strings:")
>>> print(test(strs))
>>> strs =  ['cat', 'dog', 'shatter', 'donut', 'at', 'todo', '']
>>> print("\nOriginal strings:")
>>> print(strs)
>>> print("Longest string of the said list of strings:")
>>> print(test(strs))
```

8.Write a Python program to create string consisting of the non-negative integers up to n inclusive.

Solution:

```
>>> def test(n):
>>>     return ' '.join(map(str,range(n+1)))
>>> n = 4
>>> print("Non-negative integer:")
>>> print(n)
>>> print("Non-negative integers up to n inclusive:")
>>> print(test(n))
>>> n = 15
>>> print("\nNon-negative integer:")
>>> print(n)
>>> print("Non-negative integers up to n inclusive:")
>>> print(test(n))
```

9.Write a Python program to find the indices of all occurrences of target in the uneven matrix.

Solution:

```
>>> def test(M, T):
>>>     return [[i, j] for i, row in enumerate(M) for j, n in enumerate(row) if n == T]
>>> M = [[1, 3, 2, 32, 19], [19, 2, 48, 19], [], [9, 35, 4], [3, 19]]
>>> T = 19
>>> print("Matrix:")
>>> print(M)
>>> print("Target value:")
```

```
>>> print(T)
>>> print("Indices of all occurrences of the target value in the said uneven matrix:")
print(test(M,T))
>>>
>>> M = [[1, 2, 3, 2], [], [7, 9, 2, 1, 4]]
>>> T = 2
>>> print("\nMatrix:")
>>> print(M)
>>> print("Target value:")
>>> print(T)
>>> print("Indices of all occurrences of the target value in the said uneven matrix:")
>>> print(test(M,T))
```

10.Write a Python program to split a given string (s) into strings if there is a space in the string, otherwise split on commas if there is a comma, otherwise return the list of lowercase letters with odd order (order of a = 0, b = 1, etc.)

Solution:

```
>>> def test(s):
>>>     if " " in s:
>>>         return s.split(" ")
>>>     if "," in s:
>>>         return s.split(",")
>>>     return [c for c in s if c.islower() and ord(c) % 2 == 0]
>>>
>>> strs = "a b c d"
>>> print("Original string:")
>>> print(strs)
>>> print("Split the said string into strings if there is a space in the string, \n otherwise split on commas if there is a comma, \n otherwise return the list of lowercase letters with odd order:")
>>> print(test(strs))
>>> strs = "a,b,c,d"
>>> print("\nOriginal string:")
>>> print(strs)
>>> print("Split the said string into strings if there is a space in the string, \n otherwise split on commas if there is a comma, \n otherwise return the list of lowercase letters with odd order:")
>>> print(test(strs))
>>> strs = "abcd"
>>> print("\nOriginal string:")
>>> print(strs)
```

```
>>> print("Split the said string into strings if there is a space in the string, \n otherwise split on commas if there is a comma, \n otherwise return the list of lowercase letters with odd order:")
>>> print(test(strs))
```