**Python Programs**

**1. Write a Python program to find a list of integers with exactly two occurrences of nineteen and at least three occurrences of five. Return True otherwise False.**

**Input:**

[19, 19, 15, 5, 3, 5, 5, 2]

**Output:**

True

**Input:**

[19, 15, 15, 5, 3, 3, 5, 2]

**Output:**

False

**Input:**

[19, 19, 5, 5, 5, 5, 5]

**Output:**

True

2. **Write a Python program that accepts a list of integers and calculates the length and the fifth element. Return true if the length of the list is 8 and the fifth element occurs thrice in the said list.**

**Input:**

[19, 19, 15, 5, 5, 5, 1, 2]

**Output:**

True

**Input:**

[19, 15, 5, 7, 5, 5, 2]

**Output:**

False

**Input:**

[11, 12, 14, 13, 14, 13, 15, 14]

**Output:**

True

**Input:**

[19, 15, 11, 7, 5, 6, 2]

**Output:**

False

Input:

922

Output:

True

Input:

914

Output:

False

Input:

854

Output:

True

Input:

854

Output:

True

3. 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.

**Input:** 2

**Output:**

[2, 4]

Input: 10

**Output:**

[10, 12, 14, 16, 18, 20, 22, 24, 26, 28]

**Input**: 3

**Output:**

[3, 5, 7]

**Input:** 17

**Output:**

[17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]

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

**Input:**

['a', 'abb', 'sfs', 'oo', 'de', 'sfde']

**Output:**

True

**Input:**

['a', 'abb', 'sfs', 'oo', 'ee', 'sfde']

**Output:**

False

**Input:**

['a', 'abb', 'sad', 'ooaaesdfe', 'sfsdfde', 'sfsd', 'sfsdf', 'qwrew']

Output:

False

Input:

['a', 'abb', 'sad', 'ooaaesdfe', 'sfsdfde', 'sfsd', 'sfsdf', 'qwsfsdfrew']

Output:

True

5. **Write a Python program to test a list of one hundred integers between 0 and 999, which all differ by ten from one another. Return True otherwise False.**

**Input:**

[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 710, 720, 730, 740, 750, 760, 770, 780, 790, 800, 810, 820, 830, 840, 850, 860, 870, 880, 890, 900, 910, 920, 930, 940, 950, 960, 970, 980, 990]

**Output:**

True

**Input:**

[0, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440, 460, 480, 500, 520, 540, 560, 580, 600, 620, 640, 660, 680, 700, 720, 740, 760, 780, 800, 820, 840, 860, 880, 900, 920, 940, 960, 980]

**Output:**

False

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

**Input:**

[0, 1, 2, 3, 4, 5]

**Output:**

False

**Input:**

[1, 1, 1, 1, 1, 1]

**Output:**

True

**Input:**

[2, 2, 2, 2, 2]

**Output:**

False

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

**Input:** Python, Exercises.

**Output:**

[[ 'Python', 'Exercises.'], [' ', ', ']]

Input: The dance, held in the school gym, ended at midnight.

Output:

[['The', 'dance', 'held', 'in', 'the', 'school', 'gym', 'ended', 'at', 'midnight.'], [' ', ', ', ' ', ' ', ' ', ' ', ', ', ' ', ' ']]

Input: The colors in my studyroom are blue, green, and yellow.

Output:

[['The', 'colors', 'in', 'my', 'studyroom', 'are', 'blue', 'green', 'and', 'yellow.'], [' ', ' ', ' ', ' ', ' ', ' ', ', ', ', ', ' ']]

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

**Input:**

[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]

**Output:**

True

**Input:**

[1, 2, 3, 3, 1, 2, 3, 3, 1, 2, 3, 3, 1, 2, 3, 3]

Output:

False

**Input:**

[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]

**Output:**

False

10. **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.**

**Input:**

( ()) ((()()())) (()) ()

**Output:**

['(())', '((()()()))', '(())', '()']

Input:

() (( ( )() ( )) ) ( ())

**Output:**

['()', '((()()()))', '(())']

11. **Write a Python program to find the indexes of numbers in a given list below a given threshold.**

**Original list:**

[0, 12, 45, 3, 4923, 322, 105, 29, 15, 39, 55]

Threshold: 100

Check the indexes of numbers of the said list below the given threshold:

[0, 1, 2, 3, 7, 8, 9, 10]

Original list:

[0, 12, 4, 3, 49, 9, 1, 5, 3]

Threshold: 10

Check the indexes of numbers of the said list below the given threshold:

[0, 2, 3, 5, 6, 7, 8]

12. **Write a Python program to check whether the given strings are palindromes or not. Return True otherwise False.**

**Input:**

['palindrome', 'madamimadam', '', 'foo', 'eyes']

**Output:**

[False, True, True, False, False]

13. **Write a Python program to find strings in a given list starting with a given prefix.**

**Input:**

[( ca,('cat', 'car', 'fear', 'center'))]

**Output:**

['cat', 'car']

**Input:**

[(do,('cat', 'dog', 'shatter', 'donut', 'at', 'todo'))]

**Output:**

['dog', 'donut']

14. **Write a Python program to find the length of a given list of non-empty strings.**

**Input:**

['cat', 'car', 'fear', 'center']

**Output:**

[3, 3, 4, 6]

**Input:**

['cat', 'dog', 'shatter', 'donut', 'at', 'todo', '']

**Output:**

[3, 3, 7, 5, 2, 4, 0]

15.**Write a Python program to find the longest string in a given list of strings.**

**Input:**

['cat', 'car', 'fear', 'center']

**Output:**

center

**Input:**

['cat', 'dog', 'shatter', 'donut', 'at', 'todo', '']

**Output:**

Shatter

16. **Write a Python program to find strings in a given list containing a given substring.**

**Input:**

[(ca,('cat', 'car', 'fear', 'center'))]

**Output:**

['cat', 'car']

**Input:**

[(o,('cat', 'dog', 'shatter', 'donut', 'at', 'todo', ''))]

**Output:**

['dog', 'donut', 'todo']

Input:

[(oe,('cat', 'dog', 'shatter', 'donut', 'at', 'todo', ''))]

Output:

[]

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

**Input:**

4

**Output:**

0 1 2 3 4

**Input:**

15

**Output:**

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

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

**Input:**

a b c d

Split the said string into strings if there is a space in the string,

otherwise split on commas if there is a comma,

**Output:**

['a', 'b', 'c', 'd']

**Input:**

a,b,c,d

Split the said string into strings if there is a space in the string,

otherwise split on commas if there is a comma,

**Output:**

['a', 'b', 'c', 'd']

**Input:**

abcd

Split the said string into strings if there is a space in the string,

otherwise split on commas if there is a comma,

**Output:**

['b', 'd']

**18. Write a Python program to determine the direction ('increasing' or 'decreasing') of monotonic sequence numbers.**

**Input:**

[1, 2, 3, 4, 5, 6]

**Output:**

Increasing.

**Input:**

[6, 5, 4, 3, 2, 1]

**Output:**

Decreasing.

**Input:**

[19, 19, 5, 5, 5, 5, 5]

Output:

Not a monotonic sequence!

**19. Write a Python program to check, for each string in a given list, whether the last character is an isolated letter or not. Return True otherwise False.**

**Input:**

['cat', 'car', 'fear', 'center']

**Output:**

[False, False, False, False]

**Input:**

['ca t', 'car', 'fea r', 'cente r']

**Output:**

[True, False, True, True]

20. **Write a Python program to compute the sum of the ASCII values of the upper-case characters in a given string.**

**Input:**

PytHon ExerciSEs

**Output:**

373

**Input:**

JavaScript

**Output:**

157

**21. Write a Python program to find the indices at which the numbers in the list drop.**

**NOTE: You can detect multiple drops just by checking if nums[i] < nums[i-1]**

**Input:**

[0, -1, 3, 8, 5, 9, 8, 14, 2, 4, 3, -10, 10, 17, 41, 22, -4, -4, -15, 0]

**Output:**

[1, 4, 6, 8, 10, 11, 15, 16, 18]

**Input:**

[6, 5, 4, 3, 2, 1]

**Output:**

[1, 2, 3, 4, 5]

**Input:**

[1, 19, 5, 15, 5, 25, 5]

Output:

[0, 2, 4, 6]

**22. Write a Python program to create a list whose ith element is the maximum of the first i elements from an input list.**

**Input:**

[0, -1, 3, 8, 5, 9, 8, 14, 2, 4, 3, -10, 10, 17, 41, 22, -4, -4, -15, 0]

**Output:**

[0, 0, 3, 8, 8, 9, 9, 14, 14, 14, 14, 14, 14, 17, 41, 41, 41, 41, 41, 41]

**Input:**

[6, 5, 4, 3, 2, 1]

**Output:**

[6, 6, 6, 6, 6, 6]

**Input:**

**[1, 19, 5, 15, 5, 25, 5]**

**Output:**

[1, 19, 19, 19, 19, 25, 25]

**23. Write a Python program to select a string from a given list of strings with the most unique characters.**

**Input:**

['cat', 'catatatatctsa', 'abcdefhijklmnop', '124259239185125', '', 'foo', 'unique']

**Output:**

abcdefhijklmnop

**Input:**

['Green', 'Red', 'Orange', 'Yellow', '', 'White']

Output:

Orange

**24. Write a Python program to find a list of strings that have fewer total characters (including repetitions).**

**Input:**

[['this', 'list', 'is', 'narrow'], ['I', 'am', 'shorter but wider']]

**Output:**

['this', 'list', 'is', 'narrow']

**Input:**

[['Red', 'Black', 'Pink'], ['Green', 'Red', 'White']]

**Output:**

['Red', 'Black', 'Pink']

**25. Write a python program to get the following output**

**Output**:

Kelly Kelly Kelly Kelly Kelly

Jessa Jessa Jessa Jessa Jessa

Emma Emma Emma Emma Emma