1. WAP in python to sum all the items in a list.

2. WAP in python to multiply all the items in a list.

3. WAP in python to get the largest number from a list.

4. WAP in python to get the smallest number from a list.

5. WAP in python to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

Sample List : ['abc', 'xyz', 'aba', '1221']

Expected Result : 2

6. WAP in python to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.

Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]

Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

7. WAP in python to remove duplicates from a list.

8. WAP in python to check a list is empty or not.

9. WAP in python to clone or copy a list.

10. WAP in python to find the list of words that are longer than n from a given list of words.

11. Write a Python function that takes two lists and returns True if they have at least one common member.

12. WAP in python to print a specified list after removing the 0th, 4th and 5th elements.

Sample List : ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

Expected Output : ['Green', 'White', 'Black']

13. WAP in python to generate a 3*4*6 3D array whose each element is *.

14. WAP in python to print the numbers of a specified list after removing even numbers from it.

15. WAP in python to shuffle and print a specified list.

16. WAP in python to generate and print a list of first and last 5 elements where the values are square of numbers between 1 and 30 (both included).

17. WAP in python to generate and print a list except for the first 5 elements, where the values are square of numbers between 1 and 30 (both included).

18. WAP in python to generate all permutations of a list in Python.

19. WAP in python to get the difference between the two lists.

20. WAP in python access the index of a list.

21. WAP in python to convert a list of characters into a string.

22. WAP in python to find the index of an item in a specified list.

23. WAP in python to flatten a shallow list.

24. WAP in python to append a list to the second list.

25. WAP in python to select an item randomly from a list.

26. WAP in python to check whether two lists are circularly identical.

27. WAP in python to find the second smallest number in a list.

28. WAP in python to find the second largest number in a list.

29. WAP in python to get unique values from a list.

30. WAP in python to get the frequency of the elements in a list.

31. WAP in python to count the number of elements in a list within a specified range.

32. WAP in python to check whether a list contains a sublist.

33. WAP in python to generate all sublists of a list.

34. WAP in python using Sieve of Eratosthenes method for computing primes upto a specified number.

Note: In mathematics, the sieve of Eratosthenes, (Ancient Greek: κόσκινον Ἐρατοσθένους, kóskinon Eratosthénous) one of a number of prime number sieves, is a simple, ancient algorithm for finding all prime numbers up to any given limit.

35. WAP in python to create a list by concatenating a given list which range goes from 1 to n.

Sample list : ['p', 'q']

n =5

Sample Output : ['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5']

36. WAP in python to get variable unique identification number or string.

37. WAP in python to find common items from two lists.

38. WAP in python to change the position of every n-th value with the (n+1)th in a list.

Sample list: [0,1,2,3,4,5]

Expected Output: [1, 0, 3, 2, 5, 4]

39. WAP in python to convert a list of multiple integers into a single integer.

Sample list: [11, 33, 50]

Expected Output: 113350

40. WAP in python to split a list based on first character of word.

41. WAP in python to create multiple lists.

42. WAP in python to find missing and additional values in two lists.

Sample data : Missing values in second list: b,a,c

Additional values in second list: g,h

43. WAP in python to split a list into different variables.

44. WAP in python to generate groups of five consecutive numbers in a list.

45. WAP in python to convert a pair of values into a sorted unique array.

46. WAP in python to select the odd items of a list.

47. WAP in python to insert an element before each element of a list.

48. WAP in python to print a nested lists (each list on a new line) using the print() function.

49. WAP in python to convert list to list of dictionaries.

Sample lists: ["Black", "Red", "Maroon", "Yellow"], ["#000000", "#FF0000", "#800000", "#FFFF00"]

Expected Output: [{'color_name': 'Black', 'color_code': '#000000'}, {'color_name': 'Red', 'color_code': '#FF0000'}, {'color_name': 'Maroon', 'color_code': '#800000'}, {'color_name': 'Yellow', 'color_code': '#FFFF00'}]

50. WAP in python to sort a list of nested dictionaries.

51. WAP in python to split a list every Nth element.

Sample list: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']

Expected Output: [['a', 'd', 'g', 'j', 'm'], ['b', 'e', 'h', 'k', 'n'], ['c', 'f', 'i', 'l']]

52. WAP in python to compute the difference between two lists.

Sample data: ["red", "orange", "green", "blue", "white"], ["black", "yellow", "green", "blue"]

Expected Output:

Color1-Color2: ['white', 'orange', 'red']

Color2-Color1: ['black', 'yellow']

53. WAP in python to create a list with infinite elements.

54. WAP in python to concatenate elements of a list.

55. WAP in python to remove key values pairs from a list of dictionaries.

56. WAP in python to convert a string to a list.

57. WAP in python to check if all items of a given list of strings is equal to a given string.

58. WAP in python to replace the last element in a list with another list.

Sample data : [1, 3, 5, 7, 9, 10], [2, 4, 6, 8]

Expected Output: [1, 3, 5, 7, 9, 2, 4, 6, 8]

59. WAP in python to check whether the n-th element exists in a given list.

60. WAP in python to find a tuple, the smallest second index value from a list of tuples.

61. WAP in python to create a list of empty dictionaries.

62. WAP in python to print a list of space-separated elements.

63. WAP in python to insert a given string at the beginning of all items in a list.

Sample list : [1,2,3,4], string : emp

Expected output : ['emp1', 'emp2', 'emp3', 'emp4']

64. WAP in python to iterate over two lists simultaneously.

65. WAP in python to move all zero digits to end of a given list of numbers.

Expected output:

Original list:

[3, 4, 0, 0, 0, 6, 2, 0, 6, 7, 6, 0, 0, 0, 9, 10, 7, 4, 4, 5, 3, 0, 0, 2, 9, 7, 1]

Move all zero digits to end of the said list of numbers:

[3, 4, 6, 2, 6, 7, 6, 9, 10, 7, 4, 4, 5, 3, 2, 9, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

66. WAP in python to find the list in a list of lists whose sum of elements is the highest.

Sample lists: [1,2,3], [4,5,6], [10,11,12], [7,8,9]

Expected Output: [10, 11, 12]

67. WAP in python to find all the values in a list are greater than a specified number.

68. WAP in python to extend a list without append.

Sample data: [10, 20, 30]

[40, 50, 60]

Expected output : [40, 50, 60, 10, 20, 30]

69. WAP in python to remove duplicates from a list of lists.

Sample list : [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]

New List : [[10, 20], [30, 56, 25], [33], [40]]

70. WAP in python to find the items starts with specific character from a given list.

Expected Output:

Original list:

['abcd', 'abc', 'bcd', 'bkie', 'cder', 'cdsw', 'sdfsd', 'dagfa', 'acjd']

Items start with a from the said list:

['abcd', 'abc', 'acjd']

Items start with d from the said list:

['dagfa']

Items start with w from the said list:

[]

71. WAP in python to check whether all dictionaries in a list are empty or not.

Sample list : [{},{},{}]

Return value : True

Sample list : [{1,2},{},{}]

Return value : False

72. WAP in python to flatten a given nested list structure.

Original list: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]

Flatten list:

[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

73. WAP in python to remove consecutive duplicates of a given list.

Original list:

[0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

After removing consecutive duplicates:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 4]

74. WAP in python to pack consecutive duplicates of a given list elements into sublists.

Original list:

[0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

After packing consecutive duplicates of the said list elements into sublists:

[[0, 0], [1], [2], [3], [4, 4], [5], [6, 6, 6], [7], [8], [9], [4, 4]]

75. WAP in python to create a list reflecting the run-length encoding from a given list of integers or a given list of characters.

Original list:

[1, 1, 2, 3, 4, 4.3, 5, 1]

List reflecting the run-length encoding from the said list:

[[2, 1], [1, 2], [1, 3], [1, 4], [1, 4.3], [1, 5], [1, 1]]

Original String:

automatically

List reflecting the run-length encoding from the said string:

[[1, 'a'], [1, 'u'], [1, 't'], [1, 'o'], [1, 'm'], [1, 'a'], [1, 't'], [1, 'i'], [1, 'c'], [1, 'a'], [2, 'l'], [1, 'y']]

76. WAP in python to create a list reflecting the modified run-length encoding from a given list of integers or a given list of characters.

Original list:

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

List reflecting the modified run-length encoding from the said list:

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

Original String:

aabcddddadnss

List reflecting the modified run-length encoding from the said string:

[[2, 'a'], 'b', 'c', [4, 'd'], 'a', 'd', 'n', [2, 's']]

77. WAP in python to decode a run-length encoded given list.

Original encoded list:

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

Decode a run-length encoded said list:

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

78. WAP in python to split a given list into two parts where the length of the first part of the list is given.

Original list:

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

Length of the first part of the list: 3

Splited the said list into two parts:

([1, 1, 2], [3, 4, 4, 5, 1])

79. WAP in python to remove the K'th element from a given list, print the new list.

Original list:

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

After removing an element at the kth position of the said list:

[1, 1, 3, 4, 4, 5, 1]

80. WAP in python to insert an element at a specified position into a given list.

Original list:

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

After inserting an element at kth position in the said list:

[1, 1, 12, 2, 3, 4, 4, 5, 1]

81. WAP in python to extract a given number of randomly selected elements from a given list.

Original list:

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

Selected 3 random numbers of the above list:

[4, 4, 1]

82. WAP in python to generate the combinations of n distinct objects taken from the elements of a given list.

Original list: [1, 2, 3, 4, 5, 6, 7, 8, 9] Combinations of 2 distinct objects: [1, 2] [1, 3] [1, 4] [1, 5] .... [7, 8] [7, 9] [8, 9]

83. WAP in python to round every number of a given list of numbers and print the total sum multiplied by the length of the list.

Original list: [22.4, 4.0, -16.22, -9.1, 11.0, -12.22, 14.2, -5.2, 17.5]

Result:

243

84. WAP in python to round the numbers of a given list, print the minimum and maximum numbers and multiply the numbers by 5. Print the unique numbers in ascending order separated by space.

Original list: [22.4, 4.0, 16.22, 9.1, 11.0, 12.22, 14.2, 5.2, 17.5]

Minimum value: 4

Maximum value: 22

Result:

```
20 25 45 55 60 70 80 90 110
```

85. WAP in python to create a multidimensional list (lists of lists) with zeros.

```
Multidimensional list: [[0, 0], [0, 0], [0, 0]]
```

86. WAP in python to create a 3X3 grid with numbers.

```
3X3 grid with numbers:

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

87. WAP in python to read a matrix from console and print the sum for each column. Accept matrix rows, columns and elements for each column separated with a space(for every row) as input from the user.

```
Input rows: 2

Input columns: 2

Input number of elements in a row (1, 2, 3):

1 2

3 4

sum for each column:

4 6
```

88. WAP in python to read a square matrix from console and print the sum of matrix primary diagonal. Accept the size of the square matrix and elements for each column separated with a space (for every row) as input from the user.

```
Input the size of the matrix: 3

2 3 4

4 5 6

3 4 7

Sum of matrix primary diagonal:
```

14

89. WAP in python to Zip two given lists of lists.

Original lists:

[[1, 3], [5, 7], [9, 11]]

[[2, 4], [6, 8], [10, 12, 14]]

Zipped list:

[[1, 3, 2, 4], [5, 7, 6, 8], [9, 11, 10, 12, 14]]

90. WAP in python to count number of lists in a given list of lists.

Original list:

[[1, 3], [5, 7], [9, 11], [13, 15, 17]]

Number of lists in said list of lists:

4

Original list:

[[2, 4], [[6, 8], [4, 5, 8]], [10, 12, 14]]

Number of lists in said list of lists:

3

91. WAP in python to find the list with maximum and minimum length.

Original list:

[[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]

List with maximum length of lists:

(3, [13, 15, 17])

List with minimum length of lists:

(1, [0])

Original list:

[[0], [1, 3], [5, 7], [9, 11], [3, 5, 7]] List with maximum length of lists:

(3, [3, 5, 7])

List with minimum length of lists:

(1, [0])

Original list:

[[12], [1, 3], [1, 34, 5, 7], [9, 11], [3, 5, 7]]

List with maximum length of lists:

(4, [1, 34, 5, 7])

List with minimum length of lists:

(1, [12])

92. WAP in python to check if a nested list is a subset of another nested list.

Original list:

[[1, 3], [5, 7], [9, 11], [13, 15, 17]]

[[1, 3], [13, 15, 17]]

If the one of the said list is a subset of another.:

True

Original list:

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

[[[3, 4], [5, 6]]]

If the one of the said list is a subset of another.:

True

Original list:

[[[1, 2], [2, 3]], [[3, 4], [5, 7]]]

[[[3, 4], [5, 6]]]

If the one of the said list is a subset of another.:

False

93. WAP in python to count the number of sublists contain a particular element.

Original list:

[[1, 3], [5, 7], [1, 11], [1, 15, 7]]

Count 1 in the said list:

3

Count 7 in the said list:

2

Original list:

[['A', 'B'], ['A', 'C'], ['A', 'D', 'E'], ['B', 'C', 'D']]

Count 'A' in the said list:

3

Count 'E' in the said list:

1

94. WAP in python to count number of unique sublists within a given list.

Original list:

[[1, 3], [5, 7], [1, 3], [13, 15, 17], [5, 7], [9, 11]]

Number of unique lists of the said list:

{(1, 3): 2, (5, 7): 2, (13, 15, 17): 1, (9, 11): 1}

Original list:

[['green', 'orange'], ['black'], ['green', 'orange'], ['white']]

Number of unique lists of the said list:

{('green', 'orange'): 2, ('black',): 1, ('white',): 1}

95. WAP in python to sort each sublist of strings in a given list of lists.

Original list:

[[2], [0], [1, 3], [0, 7], [9, 11], [13, 15, 17]]

Sort the list of lists by length and value:

[[0], [2], [0, 7], [1, 3], [9, 11], [13, 15, 17]]

96. WAP in python to sort a given list of lists by length and value.

Original list:

[[2], [0], [1, 3], [0, 7], [9, 11], [13, 15, 17]]

Sort the list of lists by length and value:

[[0], [2], [0, 7], [1, 3], [9, 11], [13, 15, 17]]

97. WAP in python to remove sublists from a given list of lists, which contains an element outside a given range.

Original list:

[[2], [0], [1, 2, 3], [0, 1, 2, 3, 6, 7], [9, 11], [13, 14, 15, 17]]

After removing sublists from a given list of lists, which contains an element outside the given range:

[[13, 14, 15, 17]]

98. WAP in python to scramble the letters of string in a given list.

Original list:

['Python', 'list', 'exercises', 'practice', 'solution']

After scrambling the letters of the strings of the said list:

['tnPhyo', 'tlis', 'ecrsseiex', 'ccpitear', 'noiltuos']

99. WAP in python to find the maximum and minimum values in a given heterogeneous list.

Original list:

['Python', 3, 2, 4, 5, 'version']

Maximum and Minimum values in the said list:

(5, 2)

100. WAP in python to extract common index elements from more than one given list.

Original lists:

[1, 1, 3, 4, 5, 6, 7]

[0, 1, 2, 3, 4, 5, 7]

[0, 1, 2, 3, 4, 5, 7]

Common index elements of the said lists:

[1, 7]

101. WAP in python to sort a given matrix in ascending order according to the sum of its rows.

Original Matrix:

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

Sort the said matrix in ascending order according to the sum of its rows

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

Original Matrix:

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

Sort the said matrix in ascending order according to the sum of its rows

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

102. WAP in python to extract specified size of strings from a give list of string values.

Original list:

['Python', 'list', 'exercises', 'practice', 'solution']

length of the string to extract:

8

After extracting strings of specified length from the said list:

['practice', 'solution']

103. WAP in python to extract specified number of elements from a given list, which follows each other continuously.

Original list:

[1, 1, 3, 4, 4, 5, 6, 7]

Extract 2 number of elements from the said list which follows each other continuously:

[1, 4]

Original lists:

[0, 1, 2, 3, 4, 4, 4, 4, 5, 7]

Extract 4 number of elements from the said list which follows each other continuously:

[4]

104. WAP in python to find the difference between consecutive numbers in a given list.

Original list:

[1, 1, 3, 4, 4, 5, 6, 7]

Difference between consecutive numbers of the said list:

[0, 2, 1, 0, 1, 1, 1]

Original list:

[4, 5, 8, 9, 6, 10]

Difference between consecutive numbers of the said list:

[1, 3, 1, -3, 4]

105. WAP in python to compute average of two given lists.

Original list:

[1, 1, 3, 4, 4, 5, 6, 7]

[0, 1, 2, 3, 4, 4, 5, 7, 8]

Average of two lists:

3.823529411764706

106. WAP in python to count integer in a given mixed list.

Original list:

[1, 'abcd', 3, 1.2, 4, 'xyz', 5, 'pqr', 7, -5, -12.22]

Number of integers in the said mixed list:

6

107. WAP in python to remove a specified column from a given nested list.

Original Nested list:

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

After removing 1st column:

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

Original Nested list:

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

After removing 3rd column:

[[1, 2], [-2, 4], [1, -1]]

108. WAP in python to extract a specified column from a given nested list.

Original Nested list:

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

Extract 1st column:

[1, 2, 1]

Original Nested list:

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

Extract 3rd column:

[3, -5, 1]

109. WAP in python to rotate a given list by specified number of items to the right or left direction.

original List:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Rotate the said list in left direction by 4:

[4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4]

Rotate the said list in left direction by 2:

[3, 4, 5, 6, 7, 8, 9, 10, 1, 2]

Rotate the said list in Right direction by 4:

[8, 9, 10, 1, 2, 3, 4, 5, 6]

Rotate the said list in Right direction by 2:

[9, 10, 1, 2, 3, 4, 5, 6, 7, 8]

110. WAP in python to find the item with maximum occurrences in a given list.

Original list:

[2, 3, 8, 4, 7, 9, 8, 2, 6, 5, 1, 6, 1, 2, 3, 4, 6, 9, 1, 2]

Item with maximum occurrences of the said list:

2

111. WAP in python to access multiple elements of specified index from a given list.

Original list:

[2, 3, 8, 4, 7, 9, 8, 2, 6, 5, 1, 6, 1, 2, 3, 4, 6, 9, 1, 2]

Index list:

[0, 3, 5, 7, 10]

Items with specified index of the said list:

[2, 4, 9, 2, 1]

112. WAP in python to check whether a specified list is sorted or not.

Original list:

[1, 2, 4, 6, 8, 10, 12, 14, 16, 17]

Is the said list is sorted!

True

Original list:

[1, 2, 4, 6, 8, 10, 12, 14, 16, 17]

Is the said list is sorted!

False

113. WAP in python to remove duplicate dictionary from a given list.

Original list with duplicate dictionary:

[{'Green': '#008000'}, {'Black': '#000000'}, {'Blue': '#0000FF'}, {'Green': '#008000'}]

After removing duplicate dictionary of the said list:

[{'Black': '#000000'}, {'Blue': '#0000FF'}, {'Green': '#008000'}]

114. WAP in python to extract the nth element from a given list of tuples.

Original list:

[('Greyson Fulton', 98, 99), ('Brady Kent', 97, 96), ('Wyatt Knott', 91, 94), ('Beau Turnbull', 94, 98)]

Extract nth element (n = 0) from the said list of tuples:

['Greyson Fulton', 'Brady Kent', 'Wyatt Knott', 'Beau Turnbull']

Extract nth element (n = 2) from the said list of tuples:

[99, 96, 94, 98]

115. WAP in python to check if the elements of a given list are unique or not.

Original list:

[1, 2, 4, 6, 8, 2, 1, 4, 10, 12, 14, 12, 16, 17]

Is the said list contains all unique elements!

False

Original list:

[2, 4, 6, 8, 10, 12, 14]

Is the said list contains all unique elements!

True

116. WAP in python to sort a list of lists by a given index of the inner list.

Original list:

[('Greyson Fulton', 98, 99), ('Brady Kent', 97, 96), ('Wyatt Knott', 91, 94), ('Beau Turnbull', 94, 98)]

Sort the said list of lists by a given index (Index = 0) of the inner list

[('Beau Turnbull', 94, 98), ('Brady Kent', 97, 96), ('Greyson Fulton', 98, 99), ('Wyatt Knott', 91, 94)]

Sort the said list of lists by a given index (Index = 2) of the inner list

[('Wyatt Knott', 91, 94), ('Brady Kent', 97, 96), ('Beau Turnbull', 94, 98), ('Greyson Fulton', 98, 99)]

117. WAP in python to remove all elements from a given list present in another list.

Original lists:

list1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

list2: [2, 4, 6, 8]

Remove all elements from 'list1' present in 'list2:

[1, 3, 5, 7, 9, 10]

118. WAP in python to find the difference between elements (n+1th - nth) of a given list of numeric values.

Original list:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Dfference between elements (n+1th - nth) of the said list :

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

Original list:

[2, 4, 6, 8]

Dfference between elements (n+1th - nth) of the said list :

[2, 2, 2]

119. WAP in python to check if a substring presents in a given list of string values.

Original list:

['red', 'black', 'white', 'green', 'orange']

Substring to search:

ack

Check if a substring presents in the said list of string values:

True

Substring to search:

abc

Check if a substring presents in the said list of string values:

False

120. WAP in python to create a list taking alternate elements from a given list.

Original list:

['red', 'black', 'white', 'green', 'orange']

List with alternate elements from the said list:

['red', 'white', 'orange']

Original list:

[2, 0, 3, 4, 0, 2, 8, 3, 4, 2]

List with alternate elements from the said list:

[2, 3, 0, 8, 4]

121. WAP in python to find the nested lists elements which are present in another list.

Original lists:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

[[12, 18, 23, 25, 45], [7, 11, 19, 24, 28], [1, 5, 8, 18, 15, 16]]

Intersection of said nested lists:

[[12], [7, 11], [1, 5, 8]]

122. WAP in python to find common element(s) in a given nested lists.

Original lists:

[[12, 18, 23, 25, 45], [7, 12, 18, 24, 28], [1, 5, 8, 12, 15, 16, 18]]

Common element(s) in nested lists:

[18, 12]

123. WAP in python to reverse strings in a given list of string values.

Original lists:

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

Reverse strings of the said given list:

['deR', 'neerG', 'eulB', 'etihW', 'kcalB']

124. WAP in python to find the maximum and minimum product from the pairs of tuple within a given list.

The original list, tuple :

[(2, 7), (2, 6), (1, 8), (4, 9)]

Maximum and minimum product from the pairs of the said tuple of list:

(36, 8)

125. WAP in python to calculate the product of the unique numbers of a given list.

Original List : [10, 20, 30, 40, 20, 50, 60, 40]

Product of the unique numbers of the said list: 720000000

126. WAP in python to interleave multiple lists of the same length.

Original list:

list1: [1, 2, 3, 4, 5, 6, 7]

list2: [10, 20, 30, 40, 50, 60, 70]

list3: [100, 200, 300, 400, 500, 600, 700]

Interleave multiple lists:

[1, 10, 100, 2, 20, 200, 3, 30, 300, 4, 40, 400, 5, 50, 500, 6, 60, 600, 7, 70, 700]

127. WAP in python to remove words from a given list of strings containing a character or string.

Original list:

list1: ['Red color', 'Orange#', 'Green', 'Orange @', 'White']

Character list:

['#', 'color', '@']

New list:

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

128. WAP in python to calculate the sum of the numbers in a list between the indices of a specified range.

Original list:

[2, 1, 5, 6, 8, 3, 4, 9, 10, 11, 8, 12]

Range: 8 , 10

Sum of the specified range:

29

129. WAP in python to reverse each list in a given list of lists.

Original list of lists:

[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]

Reverse each list in the said list of lists:

[[4, 3, 2, 1], [8, 7, 6, 5], [12, 11, 10, 9], [16, 15, 14, 13]]

130. WAP in python to count the same pair in three given lists.

Original lists:

[1, 2, 3, 4, 5, 6, 7, 8]

[2, 2, 3, 1, 2, 6, 7, 9]

[2, 1, 3, 1, 2, 6, 7, 9]

Number of same pair of the said three given lists:

3

131. WAP in python to count the frequency of consecutive duplicate elements in a given list of numbers.

Original lists:

[1, 2, 2, 2, 4, 4, 4, 5, 5, 5, 5]

Consecutive duplicate elements and their frequency:

([1, 2, 4, 5], [1, 3, 3, 4])

132. WAP in python to find all index positions of the maximum and minimum values in a given list of numbers.

Original list:

[12, 33, 23, 10, 67, 89, 45, 667, 23, 12, 11, 10, 54]

Index positions of the maximum value of the said list:

7

Index positions of the minimum value of the said list:

3

133. WAP in python to check common elements between two given list are in same order or not.

Original lists:

['red', 'green', 'black', 'orange']

['red', 'pink', 'green', 'white', 'black']

['white', 'orange', 'pink', 'black']

Test common elements between color1 and color2 are in same order?

True

Test common elements between color1 and color3 are in same order?

False

Test common elements between color2 and color3 are in same order?

False

134. WAP in python to find the difference between two list including duplicate elements.

Original lists:

[1, 1, 2, 3, 3, 4, 4, 5, 6, 7]

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

Difference between two said list including duplicate elements):

[3, 3, 4, 7]

135. WAP in python to iterate over all pairs of consecutive items in a given list.

Original lists:

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

Iterate over all pairs of consecutive items of the said list:

[(1, 1), (1, 2), (2, 3), (3, 3), (3, 4), (4, 4), (4, 5)]

136. WAP in python to remove duplicate words from a given list of strings.

Original String:

['Python', 'Exercises', 'Practice', 'Solution', 'Exercises']

After removing duplicate words from the said list of strings:

['Python', 'Exercises', 'Practice', 'Solution']

137. WAP in python to find a first even and odd number in a given list of numbers.

Original list:

[1, 3, 5, 7, 4, 1, 6, 8]

First even and odd number of the said list of numbers:

(4, 1)

138. WAP in python to sort a given mixed list of integers and strings. Numbers must be sorted before strings.

Original list:

[19, 'red', 12, 'green', 'blue', 10, 'white', 'green', 1]

Sort the said mixed list of integers and strings:

[1, 10, 12, 19, 'blue', 'green', 'green', 'red', 'white']

139. WAP in python to sort a given list of strings(numbers) numerically.

Original list:

['4', '12', '45', '7', '0', '100', '200', '-12', '-500']

Sort the said list of strings(numbers) numerically:

[-500, -12, 0, 4, 7, 12, 45, 100, 200]

140. WAP in python to remove the specific item from a given list of lists.

Original list of lists:

[['Red', 'Maroon', 'Yellow', 'Olive'], ['#FF0000', '#800000', '#FFFF00', '#808000'], ['rgb(255,0,0)', 'rgb(128,0,0)', 'rgb(255,255,0)', 'rgb(128,128,0)']]

Remove 1st list from the saod given list of lists:

[['Maroon', 'Yellow', 'Olive'], ['#800000', '#FFFF00', '#808000'], ['rgb(128,0,0)', 'rgb(255,255,0)', 'rgb(128,128,0)']]

Remove 2nd list from the saod given list of lists:

[['Red', 'Yellow', 'Olive'], ['#FF0000', '#FFFF00', '#808000'], ['rgb(255,0,0)', 'rgb(255,255,0)', 'rgb(128,128,0)']]

Remove 4th list from the saod given list of lists:

[['Red', 'Maroon', 'Yellow'], ['#FF0000', '#800000', '#FFFF00'], ['rgb(255,0,0)', 'rgb(128,0,0)', 'rgb(255,255,0)']]

141. WAP in python to remove empty lists from a given list of lists.

Original list:

[[], [], [], 'Red', 'Green', [1, 2], 'Blue', [], []]

After deleting the empty lists from the said lists of lists

['Red', 'Green', [1, 2], 'Blue']

142. WAP in python to sum a specific column of a list in a given list of lists.

Original list of lists:

[[1, 2, 3, 2], [4, 5, 6, 2], [7, 8, 9, 5]]

Sum: 1st column of the said list of lists:

12

Sum: 2nd column of the said list of lists:

15

Sum: 4th column of the said list of lists:

9

143. WAP in python to get the frequency of the elements in a given list of lists.

Original list of lists:

[[1, 2, 3, 2], [4, 5, 6, 2], [7, 8, 9, 5]]

Frequency of the elements in the said list of lists:

{1: 1, 2: 3, 3: 1, 4: 1, 5: 2, 6: 1, 7: 1, 8: 1, 9: 1}

144. WAP in python to extract every first or specified element from a given two-dimensional list.

Original list of lists:

[[1, 2, 3, 2], [4, 5, 6, 2], [7, 1, 9, 5]]

Extract every first element from the said given two dimensional list:

[1, 4, 7]

Extract every third element from the said given two dimensional list:

[3, 6, 9]

145. WAP in python to generate a number in a specified range except some specific numbers.

Generate a number in a specified range (1, 10) except [2, 9, 10]

7

Generate a number in a specified range (-5, 5) except [-5,0,4,3,2]

-4

146. WAP in python to compute the sum of digits of each number of a given list.

Original tuple:

[10, 2, 56]

Sum of digits of each number of the said list of integers:

14

Original tuple:

[10, 20, 4, 5, 'b', 70, 'a']

Sum of digits of each number of the said list of integers:

19

Original tuple:

[10, 20, -4, 5, -70]

Sum of digits of each number of the said list of integers:

19

147. WAP in python to interleave two given list into another list randomly.

Original lists:

[1, 2, 7, 8, 3, 7]

[4, 3, 8, 9, 4, 3, 8, 9]

Interleave two given list into another list randomly:

[4, 1, 2, 3, 8, 9, 4, 3, 7, 8, 9, 8, 3, 7]

148. WAP in python to remove specific words from a given list.

Original list:

['red', 'green', 'blue', 'white', 'black', 'orange']

Remove words:

['white', 'orange']

After removing the specified words from the said list:

['red', 'green', 'blue', 'black']

149. WAP in python to get all possible combinations of the elements of a given list.

Original list:

['orange', 'red', 'green', 'blue']

All possible combinations of the said list's elements:

[[], ['orange'], ['red'], ['red', 'orange'], ['green'], ['green', 'orange'], ['green', 'red'], ['green', 'red', 'orange'], ['blue'], ['blue', 'orange'], ['blue', 'red'], ['blue', 'red', 'orange'], ['blue', 'green'], ['blue', 'green', 'orange'], ['blue', 'green', 'red'], ['blue', 'green', 'red', 'orange']]

150. WAP in python to reverse a given list of lists.

Original list:

[['orange', 'red'], ['green', 'blue'], ['white', 'black', 'pink']]

Reverse said list of lists:

[['white', 'black', 'pink'], ['green', 'blue'], ['orange', 'red']]

Original list:

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

Reverse said list of lists:

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

151. WAP in python to find the maximum and minimum values in a given list within specified index range.

Original list:

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

Index range:

3 to 8

Maximum and minimum values of the said given list within index range:

(5, 0)

152. WAP in python to combine two given sorted lists using heapq module.

Original sorted lists:

[1, 3, 5, 7, 9, 11]

[0, 2, 4, 6, 8, 10]

After merging the said two sorted lists:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

153. WAP in python to check if a given element occurs at least n times in a list.

Original list:

[0, 1, 3, 5, 0, 3, 4, 5, 0, 8, 0, 3, 6, 0, 3, 1, 1, 0]

Check if 3 occurs at least 4 times in a list:

True

Check if 0 occurs at least 5 times in a list:

True

Check if 8 occurs at least 3 times in a list:

False

154. WAP in python to join two given list of lists of same length, element wise.

Original lists:

[[10, 20], [30, 40], [50, 60], [30, 20, 80]]

[[61], [12, 14, 15], [12, 13, 19, 20], [12]]

Join the said two lists element wise:

[[10, 20, 61], [30, 40, 12, 14, 15], [50, 60, 12, 13, 19, 20], [30, 20, 80, 12]]

Original lists:

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

[['p', 'q'], ['p', 's', 't'], ['u', 'v', 'w']]

Join the said two lists element wise:

[['a', 'b', 'p', 'q'], ['b', 'c', 'd', 'p', 's', 't'], ['e', 'f', 'u', 'v', 'w']]

155. WAP in python to add two given lists of different lengths, start from left.

Original lists:

[2, 4, 7, 0, 5, 8]

[3, 3, -1, 7]

Add said two lists from left:

[5, 7, 6, 7, 5, 8]

Original lists:

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

[2, 4, -3]

Add said two lists from left:

[3, 6, 0, 4, 5, 6]

156. WAP in python to add two given lists of different lengths, start from right.

Original lists:

[2, 4, 7, 0, 5, 8]

[3, 3, -1, 7]

Add said two lists from left:

[2, 4, 10, 3, 4, 15]

Original lists:

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

[2, 4, -3]

Add said two lists from left:

[1, 2, 3, 6, 9, 3]

157. WAP in python to interleave multiple given lists of different lengths.

Original lists:

[2, 4, 7, 0, 5, 8]

[2, 5, 8]

[0, 1]

[3, 3, -1, 7]

Interleave said lists of different lengths:

[2, 2, 0, 3, 4, 5, 1, 3, 7, 8, -1, 0, 7, 5, 8]

158. WAP in python to find the maximum and minimum values in a given list of tuples.

Original list with tuples:

[('V', 60), ('VI', 70), ('VII', 75), ('VIII', 72), ('IX', 78), ('X', 70)]

Maximum and minimum values of the said list of tuples:

(78, 60)

159. WAP in python to append the same value /a list multiple times to a list/list-of-lists.

Add a value(7), 5 times, to a list:

['7', '7', '7', '7', '7']

Add 5, 6 times, to a list:

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

Add a list, 4 times, to a list of lists:

[[1, 2, 5], [1, 2, 5], [1, 2, 5], [1, 2, 5]]

Add a list, 3 times, to a list of lists:

[[5, 6, 7], [1, 2, 5], [1, 2, 5], [1, 2, 5], [1, 2, 5]]

160. WAP in python to remove first specified number of elements from a given list satisfying a condition.

Remove the first 4 number of even numbers from the following list:

[3,10,4,7,5,7,8,3,3,4,5,9,3,4,9,8,5]

Output:

[3, 7, 5, 7, 3, 3, 5, 9, 3, 4, 9, 8, 5]

Original list:

[3, 10, 4, 7, 5, 7, 8, 3, 3, 4, 5, 9, 3, 4, 9, 8, 5]

Remove first 4 even numbers from the said list:

[3, 7, 5, 7, 3, 3, 5, 9, 3, 4, 9, 8, 5]

161. WAP in python to check if a given list is strictly increasing or not. Moreover, If removing only one element from the list results in a strictly increasing list, we still consider the list true.

True

True

True

True

True

True

True

True

True

True

```
True
False
False
False
False
False
```

162. WAP in python to find the last occurrence of a specified item in a given list.

```
Original list:
['s', 'd', 'f', 's', 'd', 'f', 's', 'f', 'k', 'o', 'p', 'i', 'w', 'e', 'k', 'c']
Last occurrence of f in the said list:
7
Last occurrence of c in the said list:
15
Last occurrence of k in the said list:
14
Last occurrence of w in the said list:
12
```

163. WAP in python to get the index of the first element which is greater than a specified element.

```
Original list:
[12, 45, 23, 67, 78, 90, 100, 76, 38, 62, 73, 29, 83]
Index of the first element which is greater than 73 in the said list:
4
Index of the first element which is greater than 21 in the said list:
1
Index of the first element which is greater than 80 in the said list:
5
Index of the first element which is greater than 55 in the said list:
```

3

164. WAP in python to get the items from a given list with specific condition.

Original list:

[12, 45, 23, 67, 78, 90, 45, 32, 100, 76, 38, 62, 73, 29, 83]

Number of Items of the said list which are even and greater than 45

5

165. WAP in python to split a given list into specified sized chunks.

Original list:

[12, 45, 23, 67, 78, 90, 45, 32, 100, 76, 38, 62, 73, 29, 83]

Split the said list into equal size 3

[[12, 45, 23], [67, 78, 90], [45, 32, 100], [76, 38, 62], [73, 29, 83]]

Split the said list into equal size 4

[[12, 45, 23, 67], [78, 90, 45, 32], [100, 76, 38, 62], [73, 29, 83]]

Split the said list into equal size 5

[[12, 45, 23, 67, 78], [90, 45, 32, 100, 76], [38, 62, 73, 29, 83]]

166. WAP in python to remove None value from a given list.

Original list:

[12, 0, None, 23, None, -55, 234, 89, None, 0, 6, -12]

Remove None value from the said list:

[12, 0, 23, -55, 234, 89, 0, 6, -12]

167. WAP in python to convert a given list of strings into list of lists.

Original list of strings:

['Red', 'Maroon', 'Yellow', 'Olive']

Convert the said list of strings into list of lists:

[['R', 'e', 'd'], ['M', 'a', 'r', 'o', 'o', 'n'], ['Y', 'e', 'l', 'l', 'o', 'w'], ['O', 'l', 'i', 'v', 'e']]

168. WAP in python to display vertically each element of a given list, list of lists.

Original list:

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

Display each element vertically of the said list:

a

b

c

d

e

f

Original list:

[[1, 2, 5], [4, 5, 8], [7, 3, 6]]

Display each element vertically of the said list of lists:

1 4 7

2 5 3

5 8 6

169. WAP in python to convert a given list of strings and characters to a single list of characters.

Original list:

['red', 'white', 'a', 'b', 'black', 'f']

Convert the said list of strings and characters to a single list of characters:

['r', 'e', 'd', 'w', 'h', 'i', 't', 'e', 'a', 'b', 'b', 'l', 'a', 'c', 'k', 'f']

170. WAP in python to insert an element in a given list after every nth position.

Original list:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]

Insert a in the said list after 2 nd element:

[1, 2, 'a', 3, 4, 'a', 5, 6, 'a', 7, 8, 'a', 9, 0]

Insert b in the said list after 4 th element:

[1, 2, 3, 4, 'b', 5, 6, 7, 8, 'b', 9, 0]

171. WAP in python to concatenate element-wise three given lists.

Original lists:

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

['red', 'green', 'black', 'blue', 'white']

['100', '200', '300', '400', '500']

Concatenate element-wise three said lists:

['0red100', '1green200', '2black300', '3blue400', '4white500']

172. WAP in python to remove the last N number of elements from a given list.

Original lists:

[2, 3, 9, 8, 2, 0, 39, 84, 2, 2, 34, 2, 34, 5, 3, 5]

Remove the last 3 elements from the said list:

[2, 3, 9, 8, 2, 0, 39, 84, 2, 2, 34, 2, 34]

Remove the last 5 elements from the said list:

[2, 3, 9, 8, 2, 0, 39, 84, 2, 2, 34]

Remove the last 1 element from the said list:

[2, 3, 9, 8, 2, 0, 39, 84, 2, 2, 34, 2, 34, 5, 3]

173. WAP in python to merge some list items in given list using index value.

Original lists:

['a', 'b', 'c', 'd', 'e', 'f', 'g']

Merge items from 2 to 4 in the said List:

['a', 'b', 'cd', 'e', 'f', 'g']

Merge items from 3 to 7 in the said List:

['a', 'b', 'c', 'defg']

174. WAP in python to add a number to each element in a given list of numbers.

Original lists:

[3, 8, 9, 4, 5, 0, 5, 0, 3]

Add 3 to each element in the said list:

[6, 11, 12, 7, 8, 3, 8, 3, 6]

Original lists:

[3.2, 8, 9.9, 4.2, 5, 0.1, 5, 3.11, 0]

Add 0.51 to each element in the said list:

[3.71, 8.51, 10.41, 4.71, 5.51, 0.61, 5.51, 3.62, 0.51]

175. WAP in python to find the minimum, maximum value for each tuple position in a given list of tuples.

Original list:

[(2, 3), (2, 4), (0, 6), (7, 1)]

Maximum value for each tuple position in the said list of tuples:

[7, 6]

Minimum value for each tuple position in the said list of tuples:

[0, 1]

176. WAP in python to create a new list dividing two given lists of numbers.

Original list:

[7, 2, 3, 4, 9, 2, 3]

[7, 2, 3, 4, 9, 2, 3]

[0.7777777777777778, 0.25, 1.5, 1.3333333333333333, 3.0, 2.0, 1.5]

177. WAP in python to find common elements in a given list of lists.

Original list:

[[7, 2, 3, 4, 7], [9, 2, 3, 2, 5], [8, 2, 3, 4, 4]]

Common elements of the said list of lists:

[2, 3]

Original list:

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

Common elements of the said list of lists:

['c']

178. WAP in python to insert a specified element in a given list after every nth element.

Original list:

[1, 3, 5, 7, 9, 11, 0, 2, 4, 6, 8, 10, 8, 9, 0, 4, 3, 0]

Insert 20 in said list after every 4 th element:

[1, 3, 5, 7, 20, 9, 11, 0, 2, 20, 4, 6, 8, 10, 20, 8, 9, 0, 4, 20, 3, 0]

Original list:

['s', 'd', 'f', 'j', 's', 'a', 'j', 'd', 'f', 'd']

Insert Z in said list after every 3 th element:

['s', 'd', 'f', 'Z', 'j', 's', 'a', 'Z', 'j', 'd', 'f', 'Z', 'd']

179. WAP in python to create the largest possible number using the elements of a given list of positive integers.

Original list:

[3, 40, 41, 43, 74, 9]

Largest possible number using the elements of the said list of positive integers:

9744341403

Original list:

[10, 40, 20, 30, 50, 60]

Largest possible number using the elements of the said list of positive integers:

605040302010

Original list:

[8, 4, 2, 9, 5, 6, 1, 0]

Largest possible number using the elements of the said list of positive integers:

98654210

180. WAP in python to create the smallest possible number using the elements of a given list of positive integers.

Original list:

[3, 40, 41, 43, 74, 9]

Smallest possible number using the elements of the said list of positive integers:

3404143749

Original list:

[10, 40, 20, 30, 50, 60]

Smallest possible number using the elements of the said list of positive integers:

102030405060

Original list:

[8, 4, 2, 9, 5, 6, 1, 0]

Smallest possible number using the elements of the said list of positive integers:

01245689

181. WAP in python to iterate a given list cyclically on specific index position.

Original list:

['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']

Iterate the said list cyclically on specific index position 3 :

['d', 'e', 'f', 'g', 'h', 'a', 'b', 'c']

Iterate the said list cyclically on specific index position 5 :

['f', 'g', 'h', 'a', 'b', 'c', 'd', 'e']

182. WAP in python to calculate the maximum and minimum sum of a sublist in a given list of lists.

Original list:

[[1, 2, 3, 5], [2, 3, 5, 4], [0, 5, 4, 1], [3, 7, 2, 1], [1, 2, 1, 2]]

Maximum sum of sub list of the said list of lists:

[2, 3, 5, 4]

Minimum sum of sub list of the said list of lists:

[1, 2, 1, 2]

183. WAP in python to get the unique values in a given list of lists.

Original list:

[[1, 2, 3, 5], [2, 3, 5, 4], [0, 5, 4, 1], [3, 7, 2, 1], [1, 2, 1, 2]]

Unique values of the said list of lists:

[0, 1, 2, 3, 4, 5, 7]

Original list:

[['h', 'g', 'l', 'k'], ['a', 'b', 'd', 'e', 'c'], ['j', 'i', 'y'], ['n', 'b', 'v', 'c'], ['x', 'z']]

Unique values of the said list of lists:

['e', 'd', 'c', 'b', 'x', 'k', 'n', 'h', 'g', 'j', 'i', 'a', 'l', 'y', 'v', 'z']

184. WAP in python to form Bigrams of words in a given list of strings.

From Wikipedia:

A bigram or digram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables, or words. A bigram is an n-gram for n=2. The frequency distribution of every bigram in a string is commonly used for simple statistical analysis of text in many applications, including in computational linguistics, cryptography, speech recognition, and so on.

Original list:

['Sum all the items in a list', 'Find the second smallest number in a list']

Bigram sequence of the said list:

```
[('Sum', 'all'), ('all', 'the'), ('the', 'items'), ('items', 'in'), ('in', 'a'), ('a', 'list'), ('Find', 'the'), ('the', 'second'), ('second', 'smallest'), ('smallest', 'number'), ('number', 'in'), ('in', 'a'), ('a', 'list')]
```

185. WAP in python to convert a given decimal number to binary list.

```
Original Number: 8

Decimal number ( 8 ) to binary list:

[1, 0, 0, 0]

Original Number: 45

Decimal number ( 45 ) to binary list:

[1, 0, 1, 1, 0, 1]

Original Number: 100

Decimal number ( 100 ) to binary list:

[1, 1, 0, 0, 1, 0, 0]
```

186. WAP in python to swap two sublists in a given list.

```
Original list:

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

Swap two sublists of the said list:

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

Swap two sublists of the said list:

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

187. WAP in python to convert a given list of tuples to a list of strings.

```
Original list of tuples:

[('red', 'green'), ('black', 'white'), ('orange', 'pink')]

Convert the said list of tuples to a list of strings:

['red green', 'black white', 'orange pink']

Original list of tuples:
```

[('Laiba', 'Delacruz'), ('Mali', 'Stacey', 'Drummond'), ('Raja', 'Welch'), ('Saarah', 'Stone')]

Convert the said list of tuples to a list of strings:

['Laiba Delacruz', 'Mali Stacey Drummond', 'Raja Welch', 'Saarah Stone']

188. WAP in python to sort a given list of tuples on specified element.

Original list of tuples:

[('item2', 10, 10.12), ('item3', 15, 25.1), ('item1', 11, 24.5), ('item4', 12, 22.5)]

Sort on 1st element of the tuple of the said list:

[('item1', 11, 24.5), ('item2', 10, 10.12), ('item3', 15, 25.1), ('item4', 12, 22.5)]

Sort on 2nd element of the tuple of the said list:

[('item2', 10, 10.12), ('item1', 11, 24.5), ('item4', 12, 22.5), ('item3', 15, 25.1)]

Sort on 3rd element of the tuple of the said list:

[('item2', 10, 10.12), ('item4', 12, 22.5), ('item1', 11, 24.5), ('item3', 15, 25.1)]

189. WAP in python to shift last element to first position and first element to last position in a given list.

Original list:

[1, 2, 3, 4, 5, 6, 7]

Shift last element to first position and first element to last position of the said list:

[7, 2, 3, 4, 5, 6, 1]

Original list:

['s', 'd', 'f', 'd', 's', 's', 'd', 'f']

Shift last element to first position and first element to last position of the said list:

['f', 'd', 'f', 'd', 's', 's', 'd', 's']

190. WAP in python to find the specified number of largest products from two given list, multiplying an element from each list.

Original lists:

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

[3, 6, 8, 9, 10, 6]

3 Number of largest products from the said two lists:

[60, 54, 50]

4 Number of largest products from the said two lists:

[60, 54, 50, 48]

191. WAP in python to find the maximum and minimum value of the three given lists.

Original lists:

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

[4, 3, 9, 0, 4, 3, 9]

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

Maximum value of the said three lists:

9

Minimum value of the said three lists:

0

192. WAP in python to remove all strings from a given list of tuples.

Original list:

[(100, 'Math'), (80, 'Math'), (90, 'Math'), (88, 'Science', 89), (90, 'Science', 92)]

Remove all strings from the said list of tuples:

[(100,), (80,), (90,), (88, 89), (90, 92)]

193. WAP in python to find the dimension of a given matrix.

Original list:

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

Dimension of the said matrix:

(2, 2)

Original list:

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

Dimension of the said matrix:

(2, 3)

Original list:

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

Dimension of the said matrix:

(3, 3)

194. WAP in python to sum two or more lists, the lengths of the lists may be different.

Original list:

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

Sum said lists with different lengths:

[4, 8, 8]

Original list:

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

Sum said lists with different lengths:

[8, 6, 4]

195. WAP in python to traverse a given list in reverse order, also print the elements with original index.

Original list:

['red', 'green', 'white', 'black']

Traverse the said list in reverse order:

black

white

green

red

Traverse the said list in reverse order with original index:

3 black

2 white

1 green

0 red

196. WAP in python to move a specified element in a given list.

Original list:

['red', 'green', 'white', 'black', 'orange']

Move white at the end of the said list:

['red', 'green', 'black', 'orange', 'white']

Original list:

['red', 'green', 'white', 'black', 'orange']

Move red at the end of the said list:

['green', 'white', 'black', 'orange', 'red']

Original list:

['red', 'green', 'white', 'black', 'orange']

Move black at the end of the said list:

['red', 'green', 'white', 'orange', 'black']

197. WAP in python to compute the average of nth elements in a given list of lists with different lengths.

Original list:

[[0, 1, 2], [2, 3, 4], [3, 4, 5, 6], [7, 8, 9, 10, 11], [12, 13, 14]]

Average of n-th elements in the said list of lists with different lengths:

[4.8, 5.8, 6.8, 8.0, 11.0]

198. WAP in python to compare two given lists and find the indices of the values present in both lists.

Original lists:

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

[7, 8, 5, 2, 10, 12]

Compare said two lists and get the indices of the values present in both lists:

[1, 4]

Original lists:

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

[7, 8, 5, 7, 10, 12]

Compare said two lists and get the indices of the values present in both lists:

[4]

Original lists:

[1, 2, 3, 4, 15, 6]

[7, 8, 5, 7, 10, 12]

Compare said two lists and get the indices of the values present in both lists:

[]

199. WAP in python to convert a given unicode list to a list contains strings.

Original lists:

['S001', 'S002', 'S003', 'S004']

Convert the said unicode list to a list contains strings:

['S001', 'S002', 'S003', 'S004']

200. WAP in python to pair up the consecutive elements of a given list.

Original lists:

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

Pair up the consecutive elements of the said list:

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

Original lists:

[1, 2, 3, 4, 5]

Pair up the consecutive elements of the said list:

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

201. WAP in python to check if a given string contains an element, which is present in a list.

The original string and list:

https://www.w3resource.com/python-exercises/list/

['.com', '.edu', '.tv']

Check if https://www.w3resource.com/python-exercises/list/ contains an element, which is present in the list ['.com', '.edu', '.tv']

True

The original string and list: https://www.w3resource.net

https://www.w3resource.net

['.com', '.edu', '.tv']

Check if https://www.w3resource.net contains an element, which is present in the list ['.com', '.edu', '.tv']

False

202. WAP in python to find the indexes of all None items in a given list.

Original list:

[1, None, 5, 4, None, 0, None, None]

Indexes of all None items of the list:

[1, 4, 6, 7]

203. WAP in python to join adjacent members of a given list.

Original list:

['1', '2', '3', '4', '5', '6', '7', '8']

Join adjacent members of a given list:

['12', '34', '56', '78']

Original list:

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

Join adjacent members of a given list:

['12']

204. WAP in python to check if first digit/character of each element in a given list is same or not.

Original list:

[1234, 122, 1984, 19372, 100]

Check if first digit in each element of the said given list is same or not!

True

Original list:

[1234, 922, 1984, 19372, 100]

Check if first digit in each element of the said given list is same or not!

False

Original list:

['aabc', 'abc', 'ab', 'a']

Check if first character in each element of the said given list is same or not!

True

Original list:

['aabc', 'abc', 'ab', 'ha']

Check if first character in each element of the said given list is same or not!

False

205. WAP in python to find the indices of elements of a given list, greater than a specified value.

Original list:

[1234, 1522, 1984, 19372, 1000, 2342, 7626]

Indices of elements of the said list, greater than 3000

[3, 6]

Original list:

[1234, 1522, 1984, 19372, 1000, 2342, 7626]

Indices of elements of the said list, greater than 20000

[]

206. WAP in python to remove additional spaces in a given list.

Original list:

['abc ', ' ', ' ', 'sdfds ', ' ', ' ', 'sdfds ', 'huy']

Remove additional spaces from the said list:

['abc', '', '', 'sdfds', '', '', 'sdfds', 'huy']

207. WAP in python to find the common tuples between two given lists.

Original lists:

[('red', 'green'), ('black', 'white'), ('orange', 'pink')]

[('red', 'green'), ('orange', 'pink')]

Common tuples between two said lists

[('orange', 'pink'), ('red', 'green')]

Original lists:

[('red', 'green'), ('orange', 'pink')]

[('red', 'green'), ('black', 'white'), ('orange', 'pink')]

Common tuples between two said lists

[('orange', 'pink'), ('red', 'green')]

208. Sum a list of numbers. WAP in python to sum the first number with the second and divide it by 2, then sum the second with the third and divide by 2, and so on.

Original list:

[1, 2, 3, 4, 5, 6, 7]

Sum the said list of numbers:

[1.5, 2.5, 3.5, 4.5, 5.5, 6.5]

Original list:

[0, 1, -3, 3, 7, -5, 6, 7, 11]

Sum the said list of numbers:

[0.5, -1.0, 0.0, 5.0, 1.0, 0.5, 6.5, 9.0]

209. WAP in python to count the number of groups of non-zero numbers separated by zeros of a given list of numbers.

Original list:

[3, 4, 6, 2, 0, 0, 0, 0, 0, 0, 6, 7, 6, 9, 10, 0, 0, 0, 0, 0, 5, 9, 9, 7, 4, 4, 0, 0, 0, 0, 0, 0, 5, 3, 2, 9, 7, 1]

Number of groups of non-zero numbers separated by zeros of the said list: 4

210. WAP in python to compute the sum of non-zero groups (separated by zeros) of a given list of numbers.

Original list:

[3, 4, 6, 2, 0, 0, 0, 0, 0, 0, 6, 7, 6, 9, 10, 0, 0, 0, 0, 0, 7, 4, 4, 0, 0, 0, 0, 0, 0, 5, 3, 2, 9, 7, 1, 0, 0, 0]

Compute the sum of non-zero groups (separated by zeros) of the said list of numbers: [15, 38, 15, 27]

211. WAP in python to remove an element from a given list.

Original list:

['Ricky Rivera', 98, 'Math', 90, 'Science']

After deleting an element:, using index of the element: [98, 'Math', 90, 'Science']

212. WAP in python to remove all the values except integer values from a given array of mixed values.

Original list: [34.67, 12, -94.89, 'Python', 0, 'C#']

After removing all the values except integer values from the said array of mixed values: [12, 0]

213. WAP in python to calculate the sum of two lowest negative numbers of a given array of integers.

An integer (from the Latin integer meaning "whole") is colloquially defined as a number that can be written without a fractional component. For example, 21, 4, 0, and -2048 are integers.

Original list elements: [-14, 15, -10, -11, -12, -13, 16, 17, 18, 19, 20]

Sum of two lowest negative numbers of the said array of integers: -27

Original list elements: [-4, 5, -2, 0, 3, -1, 4, 9]

Sum of two lowest negative numbers of the said array of integers: -6

214. WAP in python to sort a given positive number in descending/ascending order.

Descending -> Highest to lowest.

Ascending -> Lowest to highest

Original Number: 134543

Descending order of the said number: 544331

Ascending order of the said number: 133445

Original Number: 43750973

Descending order of the said number: 97754330

Ascending order of the said number: 3345779

215. WAP in python to merge two or more lists into a list of lists, combining elements from each of the input lists based on their positions.

Sample Output:

After merging lists into a list of lists:

[['a', 1, True], ['b', 2, False]]

[['a', 1, True], [None, 2, False]]

[['a', 1, True], ['_', 2, False]]

216. WAP in python to group the elements of a list based on the given function and returns the count of elements in each group.

Sample Output:

{6: 2, 4: 1}

{3: 2, 5: 1}

217. WAP in python to split values into two groups, based on the result of the given filtering function.

Sample Output:

```
[['white'], ['red', 'green', 'black']]
```

218. WAP in python to sort one list based on another list containing the desired indexes.

Sample Output:

```
['apples', 'bread', 'eggs', 'jam', 'milk', 'oranges']
['oranges', 'milk', 'jam', 'eggs', 'bread', 'apples']
```

219. WAP in python to build a list, using an iterator function and an initial seed value.

Sample Output:

```
[-10, -20, -30, -40]
```

220. WAP in python to map the values of a list to a dictionary using a function, where the key-value pairs consist of the original value as the key and the result of the function as the value.

Sample Output:

```
{1: 1, 2: 4, 3: 9}
```

221. WAP in python to randomize the order of the values of an list, returning a new list.

Sample Output:

```
Original list: [1, 2, 3, 4, 5, 6]
Shuffle the elements of the said list:
[3, 2, 4, 1, 6, 5]
```

222. WAP in python to get the difference between two given lists, after applying the provided function to each list element of both.

Sample Output:

```
[1.2]

[{'x': 2}]
```

223. WAP in python to create a list with the non-unique values filtered out.

Sample Output:

```
[1, 3, 5]
```

224. WAP in python to create a list with the unique values filtered out.

Sample Output:

```
[2, 4]
```

225. WAP in python to retrieve the value of the nested key indicated by the given selector list from a dictionary or list.

Sample Output:

```
Harwood

2
```

226. WAP in python to get a list of elements that exist in both lists, after applying the provided function to each list element of both.

Sample Output:

```
[2.1]
```

227. WAP in python to get the symmetric difference between two lists, after applying the provided function to each list element of both.

Sample Output:

```
[1.2, 3.4]
```

228. WAP in python to get every element that exists in any of the two given lists once, after applying the provided function to each element of both.

Sample Output:

```
[2.2, 4.1]
```

229. WAP in python to find the index of the first element in the given list that satisfies the provided testing function.

Sample Output:

```
0
```

230. WAP in python to find the indexes of all elements in the given list that satisfy the provided testing function.

Sample Output:

```
[0, 2]
```

231. WAP in python to split values into two groups, based on the result of the given filter list.

Sample Output:

```
[['red', 'green', 'pink'], ['blue']]
```

232. WAP in python to chunk a given list into smaller lists of a specified size.

Sample Output:

```
[[1, 2, 3], [4, 5, 6], [7, 8]]
```

233. WAP in python to chunk a given list into n smaller lists.

Sample Output:

```
[[1, 2], [3, 4], [5, 6], [7]]
```

234. WAP in python to convert a given number (integer) to a list of digits.

Sample Output:

```
[1, 2, 3]

[1, 3, 4, 7, 8, 2, 3]
```

235. WAP in python to find the index of the last element in the given list that satisfies the provided testing function.

Sample Output:

```
2
```

236. WAP in python to find the items that are parity outliers in a given list.

Sample Output:

```
[1, 3]

[2, 4, 6]
```

237. WAP in python to convert a given list of dictionaries into a list of values corresponding to the specified key.

Sample Output:

```
[8, 36, 34, 10]
```

238. WAP in python to calculate the average of a given list, after mapping each element to a value using the provided function.

Sample Output:

```
5.0

15.0
```

239. WAP in python to find the value of the first element in the given list that satisfies the provided testing function.

Sample Output:

```
1

2
```

240. WAP in python to find the value of the last element in the given list that satisfies the provided testing function.

Sample Output:

```
3

4
```

241. WAP in python to create a dictionary with the unique values of a given list as keys and their frequencies as the values.

Sample Output:

```
{'a': 4, 'b': 2, 'f': 2, 'c': 1, 'e': 2}

{3: 4, 4: 2, 7: 1, 5: 2, 9: 1, 0: 1, 2: 1}
```

242. WAP in python to get the symmetric difference between two iterables, without filtering out duplicate values.

Sample Output:

```
[30, 40]
```

243. WAP in python to check if a given function returns True for every element in a list.

Sample Output:

```
True

False

False
```

244. WAP in python to initialize a list containing the numbers in the specified range where start and end are inclusive and the ratio between two terms is step. Returns an error if step equals 1.

Sample Output:

```
[1, 2, 4, 8, 16, 32, 64, 128, 256]

[3, 6, 12, 24, 48, 96, 192]

[1, 4, 16, 64, 256]
```

245. WAP in python to that takes any number of iterable objects or objects with a length property and returns the longest one.

Sample Output:

```
Green

[1, 2, 3, 4, 5]

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

246. WAP in python to check if a given function returns True for at least one element in the list.

Sample Output:

```
True

False
```

247. WAP in python to calculate the difference between two iterables, without filtering duplicate values.

Sample Output:

```
[3]
```

248. WAP in python to get the maximum value of a list, after mapping each element to a value using a given function.

Sample Output:

```
8
```

249. WAP in python to get the minimum value of a list, after mapping each element to a value using a given function.

Sample Output:

```
2
```

250. WAP in python to calculate the sum of a list, after mapping each element to a value using the provided function.

Sample Output:

```
20
```

251. WAP in python to initialize and fills a list with the specified value.

Sample Output:

```
[0, 0, 0, 0, 0, 0, 0]

[3, 3, 3, 3, 3, 3, 3, 3]

[-2, -2, -2, -2, -2]

[3.2, 3.2, 3.2, 3.2, 3.2]
```

252. WAP in python to get the n maximum elements from a given list of numbers.

Sample Output:

```
Original list elements:

[1, 2, 3]
```

Maximum values of the said list: [3]

Original list elements:

[1, 2, 3]

Two maximum values of the said list: [3, 2]

Original list elements:

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

Threee maximum values of the said list: [0, -1, -2]

Original list elements:

[2.2, 2, 3.2, 4.5, 4.6, 5.2, 2.9]

Two maximum values of the said list: [5.2, 4.6]

253. WAP in python to get the n minimum elements from a given list of numbers.

Sample Output:

Original list elements:

[1, 2, 3]

Minimum values of the said list: [1]

Original list elements:

[1, 2, 3]

Two minimum values of the said list: [1, 2]

Original list elements:

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

Threee minimum values of the said list: [-5, -4, -3]

Original list elements:

[2.2, 2, 3.2, 4.5, 4.6, 5.2, 2.9]

Two minimum values of the said list: [2, 2.2]

254. WAP in python to get the weighted average of two or more numbers.

Sample Output:

Original list elements:

```
[10, 50, 40]
[2, 5, 3]
Weighted average of the said two list of numbers:
39.0
Original list elements:
[82, 90, 76, 83]
[0.2, 0.35, 0.45, 32]
Weighted average of the said two list of numbers:
82.97272727272727
```

255. WAP in python to perform a deep flattens a list.

Sample Output:

```
Original list elements:
[1, [2], [[3], [4], 5], 6]
Deep flatten the said list:
[1, 2, 3, 4, 5, 6]
Original list elements:
[[[1, 2, 3], [4, 5]], 6]
Deep flatten the said list:
[1, 2, 3, 4, 5, 6]
```

256. WAP in python to get the powerset of a given iterable.

Sample Output:

```
Original list elements:
[1, 2]
Powerset of the said list:
[(), (1,), (2,), (1, 2)]
Original list elements:
[1, 2, 3, 4]
```

```
Powerset of the said list:

[(), (1,), (2,), (3,), (4,), (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4), (1, 2, 3), (1, 2, 4), (1, 3, 4), (2, 3, 4), (1, 2, 3, 4)]
```

257. WAP in python to check if two given lists contain the same elements regardless of order.

Sample Output:

```
Original list elements:

[1, 2, 4]

[2, 4, 1]

Check two said lists contain the same elements regardless of order!

True

Original list elements:

[1, 2, 3]

[1, 2, 3]

Check two said lists contain the same elements regardless of order!

True

Original list elements:

[1, 2, 3]

[1, 2, 4]

Check two said lists contain the same elements regardless of order!

False
```

258. WAP in python to create a given flat list of all the keys in a flat dictionary.

Sample Output:

```
Original directory elements:

{'Laura': 10, 'Spencer': 11, 'Bridget': 9, 'Howard ': 10}

Flat list of all the keys of the said dictionary:

['Laura', 'Spencer', 'Bridget', 'Howard ']
```

259. WAP in python to check if a given function returns True for at least one element in the list.

Sample Output:

```
False
True
False
```

260. WAP in python to check if all the elements of a list are included in another given list.

Sample Output:

```
True
False
```

261. WAP in python to get the most frequent element in a given list of numbers.

Sample Output:

```
2
Original list:
[2, 3, 8, 4, 7, 9, 8, 2, 6, 5, 1, 6, 1, 2, 3, 2, 4, 6, 9, 1, 2]
Item with maximum frequency of the said list:
2
Original list:
[1, 2, 3, 1, 2, 3, 2, 1, 4, 3, 3]
Item with maximum frequency of the said list:
3
```

262. WAP in python to move the specified number of elements to the end of the given list.

Sample Output:

```
[4, 5, 6, 7, 8, 1, 2, 3]
[6, 7, 8, 1, 2, 3, 4, 5]
```

```
[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 3, 4, 5, 6, 7, 8]

[8, 1, 2, 3, 4, 5, 6, 7]

[2, 3, 4, 5, 6, 7, 8, 1]
```

263. WAP in python to move the specified number of elements to the start of the given list.

Sample Output:

```
[4, 5, 6, 7, 8, 1, 2, 3]

[6, 7, 8, 1, 2, 3, 4, 5]

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 3, 4, 5, 6, 7, 8]

[8, 1, 2, 3, 4, 5, 6, 7]

[2, 3, 4, 5, 6, 7, 8, 1]
```

264. WAP in python to create a two-dimensional list from given list of lists.

Sample Output:

```
[(1, 4, 7, 10), (2, 5, 8, 11), (3, 6, 9, 12)]

[(1, 4), (2, 5)]
```

265. WAP in python to generate a list, containing the Fibonacci sequence, up until the nth term.

Sample Output:

```
First 7 Fibonacci numbers:

[0, 1, 1, 2, 3, 5, 8, 13]

First 15 Fibonacci numbers:

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]

First 50 Fibonacci numbers:

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711,
28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578,
5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 165580141, 267914296,
433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049,
12586269025]
```

266. WAP in python to cast the provided value as a list if it's not one.

Sample Output:

```
<class 'list'>
[1]
<class 'tuple'>
['Red', 'Green']
<class 'set'>
['Green', 'Red']
<class 'dict'>
[1, 2, 3]
```

267. WAP in python to get the cumulative sum of the elements of a given list.

Sample Output:

```
Original list elements:
[1, 2, 3, 4]
Cumulative sum of the elements of the said list:
[1, 3, 6, 10]
Original list elements:
[-1, -2, -3, 4]
Cumulative sum of the elements of the said list:
[-1, -3, -6, -2]
```

268. WAP in python to get a list with n elements removed from the left, right.

Sample Output:

```
Original list elements:

[1, 2, 3]

Remove 1 element from left of the said list:

[2, 3]

Remove 1 element from right of the said list:

[1, 2]

Original list elements:

[1, 2, 3, 4]

Remove 2 elements from left of the said list:

[3, 4]

Remove 2 elements from right of the said list:

[1, 2]

Original list elements:

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

Remove 7 elements from left of the said list:

[2, 3, 4, 5, 6]

Remove 7 elements from right of the said list:

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

269. WAP in python to get the every nth element in a given list.

Sample Output:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[2, 4, 6, 8, 10]

[5, 10]

[6]
```

270. WAP in python to check if the elements of the first list are contained in the second one regardless of order.

Sample Output:

```
True

True

False

True
```

271. WAP in python to check if there are duplicate values in a given flat list.

Sample Output:

```
Original list:

[1, 2, 3, 4, 5, 6, 7]

Check if there are duplicate values in the said given flat list:

False

Original list:

[1, 2, 3, 3, 4, 5, 5, 6, 7]

Check if there are duplicate values in the said given flat list:

True
```