

In [10]:

```
#1

total = 0
list1 = [16, 4, 13, 9, 24]
for ele in range(0, len(list1)):
    total = total + list1[ele]
print("Sum of all elements in given list: ", total)
```

Sum of all elements in given list: 66

In [5]:

```
#2
def multiply_list(items):
    tot = 1
    for x in items:
        tot *= x
    return tot
print(multiply_list([1,4,-3]))
```

-12

In [7]:

```
#3
def max_num_in_list( list ):
    max = list[ 0 ]
    for a in list:
        if a > max:
            max = a
    return max
print(max_num_in_list([1, 3, -8, 4]))
```

4

In [9]:

```
#4
def smallest_num_in_list( list ):
    min = list[ 0 ]
    for a in list:
        if a < min:
            min = a
    return min
print(smallest_num_in_list([2,-7, 3, 1]))
```

-7

In [11]:

```
#5
def match_words(words):
    ctr = 0

    for word in words:
        if len(word) > 1 and word[0] == word[-1]:
            ctr += 1
    return ctr

print(match_words(['abc', 'xyz', 'aba', '1221']))
```

2

In [12]:

```
#6
def last(n): return n[-1]
```

```
def sort_list_last(tuples):
    return sorted(tuples, key=lambda x: x[-1])

print(sort_list_last([(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]))
```

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

In [15]:

```
#7
a = [30,20,10,20,10,50,60,50,80,50,40]

dup_items = set()
uniq_items = []
for x in a:
    if x not in dup_items:
        uniq_items.append(x)
        dup_items.add(x)

print(dup_items)
```

{40, 10, 80, 50, 20, 60, 30}

In [14]:

```
#8
l = []
if not l:
    print("List is empty")
```

List is empty

In [17]:

```
#9
original_list = [30, 22, 14, 33, 7]
new_list = list(original_list)
print(original_list)
print(new_list)
```

[30, 22, 14, 33, 7]  
[30, 22, 14, 33, 7]

In [18]:

```
#10
def long_words(n, str):
    word_len = []
    txt = str.split(" ")
    for x in txt:
        if len(x) > n:
            word_len.append(x)
    return word_len
print(long_words(3, "The quick brown fox jumps over the lazy dog"))
```

['quick', 'brown', 'jumps', 'over', 'lazy']

In [19]:

```
#11
def common_data(list1, list2):
    result = False
    for x in list1:
        for y in list2:
            if x == y:
                result = True
                return result
print(common_data([1,2,3,4,5], [5,6,7,8,9]))
print(common_data([1,2,3,4,5], [6,7,8,9]))
```

True

None

In [20]:

```
#12
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
color = [x for (i,x) in enumerate(color) if i not in (0,4,5)]
print(color)

['Green', 'White', 'Black']
```

In [21]:

```
#13
array = [[ ['*' for col in range(6)] for col in range(4)] for row in range(3)]
print(array)

[[['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'],
['*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*'], ['*', '*', '*', '*', '*', '*'],
['*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*'],
['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'],
['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'],
['*', '*', '*', '*', '*', '*']]]
```

In [23]:

```
#14
num = [7,8, 120, 25, 44, 20, 27]
num = [x for x in num if x%2!=0]
print(num)

[7, 25, 27]
```

In [24]:

```
#15
from random import shuffle
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
shuffle(color)
print(color)

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

In [25]:

```
#16
def printValues():
    l = list()
    for i in range(1,21):
        l.append(i**2)
    print(l[:5])
    print(l[-5:])

printValues()

[1, 4, 9, 16, 25]
[256, 289, 324, 361, 400]
```

In [26]:

```
#17
def printValues():
    l = list()
    for i in range(1,31):
        l.append(i**2)
    print(l[5:])
printValues()

[36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900]
```

In [27]:

```
#18
import itertools
print(list(itertools.permutations([1,2,3])))
```

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

In [28]:

```
#19
list1 = [1, 3, 5, 7, 9]
list2=[1, 2, 4, 6, 7, 8]
diff_list1_list2 = list(set(list1) - set(list2))
diff_list2_list1 = list(set(list2) - set(list1))
total_diff = diff_list1_list2 + diff_list2_list1
print(total_diff)
```

```
[9, 3, 5, 8, 2, 4, 6]
```

In [29]:

```
#20
nums = [5, 15, 35, 8, 98]
for num_index, num_val in enumerate(nums):
    print(num_index, num_val)
```

```
0 5
1 15
2 35
3 8
4 98
```

In [31]:

```
#21
s = ['p', 'q', 'r', 's']
str1 = ''.join(s)
print(str1)
```

```
pqrs
```

In [32]:

```
#22
num =[10, 30, 4, -6]
print(num.index(30))
```

```
1
```

In [33]:

```
#23
import itertools
original_list = [[2,4,3],[1,5,6], [9], [7,9,0]]
new_merged_list = list(itertools.chain(*original_list))
print(new_merged_list)
```

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

In [95]:

```
#24
list1 = [1, 2, 3, 0]
list2 = ['Orange', 'Green', 'Black']
final_list = list1 + list2
print(final_list)
```

```
[1, 2, 3, 0, 'orange', 'Green', 'Black']
```

In [96]:

```
#25
import random
```

```
color_list = ['Yellow', 'Blue', 'Green', 'White', 'Black']
print(random.choice(color_list))
```

Green

In [39]:

```
#26
list1 = [10, 10, 0, 0, 10]
list2 = [10, 10, 10, 0, 0]
list3 = [1, 10, 10, 0, 0]

print('Compare list1 and list2')
print(' '.join(map(str, list2)) in ' '.join(map(str, list1 * 2)))
print('Compare list1 and list3')
print(' '.join(map(str, list3)) in ' '.join(map(str, list1 * 2)))
```

Compare list1 and list2  
True  
Compare list1 and list3  
False

In [40]:

```
#27
def second_smallest(numbers):
    if (len(numbers)<2):
        return
    if ((len(numbers)==2) and (numbers[0] == numbers[1])):
        return
    dup_items = set()
    uniq_items = []
    for x in numbers:
        if x not in dup_items:
            uniq_items.append(x)
            dup_items.add(x)
    uniq_items.sort()
    return uniq_items[1]

print(second_smallest([1, 2, -8, -2, 0, -2]))
print(second_smallest([1, 1, 0, 0, 2, -2, -2]))
print(second_smallest([1, 1, 1, 0, 0, 0, 2, -2, -2]))
print(second_smallest([2,2]))
print(second_smallest([2]))
```

-2  
0  
0  
None  
None

In [41]:

```
#28
def second_largest(numbers):
    if (len(numbers)<2):
        return
    if ((len(numbers)==2) and (numbers[0] == numbers[1])):
        return
    dup_items = set()
    uniq_items = []
    for x in numbers:
        if x not in dup_items:
            uniq_items.append(x)
            dup_items.add(x)
    uniq_items.sort()
    return uniq_items[-2]
print(second_largest([1,2,3,4,4]))
print(second_largest([1, 1, 1, 0, 0, 0, 2, -2, -2]))
print(second_largest([2,2]))
print(second_largest([1]))
```

3  
1  
None  
None

In [42]:

```
#29
my_list = [10, 20, 30, 40, 20, 50, 60, 40]
print("Original List : ",my_list)
my_set = set(my_list)
my_new_list = list(my_set)
print("List of unique numbers : ",my_new_list)
```

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

In [43]:

```
#30
import collections
my_list = [10,10,10,10,20,20,20,20,40,40,50,50,30]
print("Original List : ",my_list)
ctr = collections.Counter(my_list)
print("Frequency of the elements in the List : ",ctr)
```

Original List : [10, 10, 10, 10, 20, 20, 20, 20, 40, 40, 50, 50, 30]  
Frequency of the elements in the List : Counter({10: 4, 20: 4, 40: 2, 50: 2, 30: 1})

In [44]:

```
#31
def count_range_in_list(li, min, max):
    ctr = 0
    for x in li:
        if min <= x <= max:
            ctr += 1
    return ctr

list1 = [10,20,30,40,40,40,70,80,99]
print(count_range_in_list(list1, 40, 100))

list2 = ['a','b','c','d','e','f']
print(count_range_in_list(list2, 'a', 'e'))
```

6  
5

In [45]:

```
#32
def is_Sublist(l, s):
    sub_set = False
    if s == []:
        sub_set = True
    elif s == l:
        sub_set = True
    elif len(s) > len(l):
        sub_set = False
    else:
        for i in range(len(l)):
            if l[i] == s[0]:
                n = 1
                while (n < len(s)) and (l[i+n] == s[n]):
                    n += 1

                if n == len(s):
                    sub_set = True

    return sub_set
```

```

a = [2,4,3,5,7]
b = [4,3]
c = [3,7]
print(is_Sublist(a, b))
print(is_Sublist(a, c))

```

True  
False

In [47]:

```

#33
from itertools import combinations
def sub_lists(my_list):
    subs = []
    for i in range(0, len(my_list)+1):
        temp = [list(x) for x in combinations(my_list, i)]
        if len(temp)>0:
            subs.extend(temp)
    return subs

```

```

l1 = [10, 20, 30, 40]
l2 = ['X', 'Y', 'Z']
print("Original list:")
print(l1)
print("S")
print(sub_lists(l1))
print("Sublists of the said list:")
print(sub_lists(l1))
print("\nOriginal list:")
print(l2)
print("Sublists of the said list:")
print(sub_lists(l2))

```

Original list:  
[10, 20, 30, 40]  
S  
[[], [10], [20], [30], [40], [10, 20], [10, 30], [10, 40], [20, 30], [20, 40], [30, 40], [10, 20, 30], [10, 20, 40], [10, 30, 40], [20, 30, 40], [10, 20, 30, 40]]  
Sublists of the said list:  
[[], [10], [20], [30], [40], [10, 20], [10, 30], [10, 40], [20, 30], [20, 40], [30, 40], [10, 20, 30], [10, 20, 40], [10, 30, 40], [20, 30, 40], [10, 20, 30, 40]]

Original list:  
['X', 'Y', 'Z']  
Sublists of the said list:  
[[], ['X'], ['Y'], ['Z'], ['X', 'Y'], ['X', 'Z'], ['Y', 'Z'], ['X', 'Y', 'Z']]

In [48]:

```

#34
def prime_eratosthenes(n):
    prime_list = []
    for i in range(2, n+1):
        if i not in prime_list:
            print(i)
            for j in range(i*i, n+1, i):
                prime_list.append(j)

print(prime_eratosthenes(100));

```

2  
3  
5  
7  
11  
13  
17  
19  
23  
29

31  
37  
41  
43  
47  
53  
59  
61  
67  
71  
73  
79  
83  
89  
97  
None

In [49]:

```
#35
my_list = ['p', 'q']
n = 4
new_list = ['{}{}'.format(x, y) for y in range(1, n+1) for x in my_list]
print(new_list)

['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4']
```

In [51]:

```
#36
x = 100
print(format(id(x), 'x'))
s = 'Yogesh'
print(format(id(s), 'x'))
```

a9d400  
7fe5d82402b0

In [52]:

```
#37
color1 = "Red", "Green", "Orange", "White"
color2 = "Black", "Green", "White", "Pink"
print(set(color1) & set(color2))

{'Green', 'White'}
```

In [53]:

```
#38
from itertools import zip_longest, chain, tee
def replace2copy(lst):
    lst1, lst2 = tee(iter(lst), 2)
    return list(chain.from_iterable(zip_longest(lst[1::2], lst[::2])))
n = [0,1,2,3,4,5]
print(replace2copy(n))
```

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

In [57]:

```
#39
L = [11,33,50]
print("Original List: ",L)
x = int("".join(map(str, L)))
print("Single Integer: ",x)
```

Original List: [11, 33, 50]  
Single Integer: 113350

In [58]:



```
#40
from itertools import groupby
from operator import itemgetter

word_list = ['be', 'have', 'do', 'say', 'get', 'make', 'go', 'know', 'take', 'see', 'come', 'think',
            'look', 'want', 'give', 'use', 'find', 'tell', 'ask', 'work', 'seem', 'feel', 'leave', 'call']

for letter, words in groupby(sorted(word_list), key=itemgetter(0)):
    print(letter)
    for word in words:
        print(word)
```

```
a
ask
b
be
c
call
come
d
do
f
feel
find
g
get
give
go
h
have
k
know
l
leave
look
m
make
s
say
see
seem
t
take
tell
think
u
use
w
want
work
```

In [59]:

```
#41
obj = {}
for i in range(1, 21):
    obj[str(i)] = []
print(obj)
```

```
{'1': [], '2': [], '3': [], '4': [], '5': [], '6': [], '7': [], '8': [], '9': [], '10': [
], '11': [], '12': [], '13': [], '14': [], '15': [], '16': [], '17': [], '18': [], '19': [
], '20': []}
```

In [61]:

```
#42
list1 = ['b', 'a', 'c', 'd', 'e', 'f']
list2 = ['d', 'e', 'f', 'g', 'h']
print('Missing values in second list: ', ', '.join(set(list1).difference(list2)))
print('Additional values in second list: ', ', '.join(set(list2).difference(list1)))
```

Missing values in second list: c,a,b  
Additional values in second list: g,h

In [62]:

```
#43
color = [("Black", "#000000", "rgb(0, 0, 0)"), ("Red", "#FF0000", "rgb(255, 0, 0)"),
         ("Yellow", "#FFFF00", "rgb(255, 255, 0)")]
var1, var2, var3 = color
print(var1)
print(var2)
print(var3)

('Black', '#000000', 'rgb(0, 0, 0)')
('Red', '#FF0000', 'rgb(255, 0, 0)')
('Yellow', '#FFFF00', 'rgb(255, 255, 0)')
```

In [63]:

```
#44
l = [[5*i + j for j in range(1,6)] for i in range(5)]
print(l)

[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15], [16, 17, 18, 19, 20], [21, 22, 23, 24, 25]]
```

In [64]:

```
#45
L = [(1, 2), (3, 4), (1, 2), (5, 6), (7, 8), (1, 2), (3, 4), (3, 4),
     (7, 8), (9, 10)]
print("Original List: ", L)
print("Sorted Unique Data:",sorted(set().union(*L)))

Original List:  [(1, 2), (3, 4), (1, 2), (5, 6), (7, 8), (1, 2), (3, 4), (3, 4), (7, 8), (9, 10)]
Sorted Unique Data: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [65]:

```
#46
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
print(x[::2])

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

In [66]:

```
#47
color = ['Red', 'Green', 'Black']
print("Original List: ",color)
color = [v for elt in color for v in ('c', elt)]
print("Original List: ",color)

Original List:  ['Red', 'Green', 'Black']
Original List:  ['c', 'Red', 'c', 'Green', 'c', 'Black']
```

In [67]:

```
#48
colors = [['Red'], ['Green'], ['Black']]
print('\n'.join([str(lst) for lst in colors]))

['Red']
['Green']
['Black']
```

In [68]:

```
#49
color_name = ["Black", "Red", "Maroon", "Yellow"]
color_code = ["#000000", "#FF0000", "#800000", "#FFFF00"]
```

```
print([{'color_name': f, 'color_code': c} for f, c in zip(color_name, color_code)])
```

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

In [69]:

```
#50
my_list = [{'key': {'subkey': 1}}, {'key': {'subkey': 10}}, {'key': {'subkey': 5}}]
print("Original List: ")
print(my_list)
my_list.sort(key=lambda e: e['key']['subkey'], reverse=True)
print("Sorted List: ")
print(my_list)
```

Original List:

```
[{'key': {'subkey': 1}}, {'key': {'subkey': 10}}, {'key': {'subkey': 5}}]
```

Sorted List:

```
[{'key': {'subkey': 10}}, {'key': {'subkey': 5}}, {'key': {'subkey': 1}}]
```

In [70]:

```
#51
C = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']
def list_slice(S, step):
    return [S[i::step] for i in range(step)]
print(list_slice(C,3))

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

In [71]:

```
#52
from collections import Counter
color1 = ["red", "orange", "green", "blue", "white"]
color2 = ["black", "yellow", "green", "blue"]
counter1 = Counter(color1)
counter2 = Counter(color2)
print("Color1-Color2: ",list(counter1 - counter2))
print("Color2-Color1: ",list(counter2 - counter1))
```

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

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

In [72]:

```
#53
import itertools
c = itertools.count()
print(next(c))
print(next(c))
print(next(c))
print(next(c))
print(next(c))
print(next(c))
```

0

1

2

3

4

In [74]:

```
#54
color = ['blue', 'green', 'orange']
print('-'.join(color))
print(''.join(color))
```

blue-green-orange

bluegreenorange

In [75]:

```
#55
original_list = [{'key1': 'value1', 'key2': 'value2'}, {'key1': 'value3', 'key2': 'value4'}]
print("Original List: ")
print(original_list)
new_list = [{k: v for k, v in d.items() if k != 'key1'} for d in original_list]
print("New List: ")
print(new_list)
```

Original List:

```
[{'key1': 'value1', 'key2': 'value2'}, {'key1': 'value3', 'key2': 'value4'}]
```

New List:

```
[{'key2': 'value2'}, {'key2': 'value4'}]
```

In [77]:

```
#56
import ast
color = ["'blue'", "'Green'", "'White'"]
print(ast.literal_eval(color))
```

```
['blue', 'Green', 'White']
```

In [79]:

```
#57
color1 = ["green", "orange", "black", "white"]
color2 = ["green", "green", "green", "green"]

print(all(c == 'red' for c in color1))
print(all(c == 'green' for c in color2))
```

False

True

In [80]:

```
#58
num1 = [1, 3, 5, 7, 9, 10]
num2 = [2, 4, 6, 8]
num1[-1:] = num2
print(num1)
```

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

In [81]:

```
#59
x = [1, 2, 3, 4, 5, 6]
xlen = len(x)-1
print(x[xlen])
```

6

In [82]:

```
#60
x = [(4, 1), (1, 2), (6, 0)]
print(min(x, key=lambda n: (n[1], -n[0])))
```

```
(6, 0)
```

In [83]:

```
#61
n = 5
l = [{} for _ in range(n)]
print(l)
```

```
[{}, {}, {}, {}, {}]
```

In [85]:

```
#62
num = [1,2,3,4,5]
print (*num)
```

1 2 3 4 5

In [86]:

```
#63
num = [1,2,3,4]
print(['emp{0}'.format(i) for i in num])
```

['emp1', 'emp2', 'emp3', 'emp4']

In [87]:

```
#64
num = [1, 2, 3]
color = ['blue', 'green', 'black']
for (a,b) in zip(num, color):
    print(a, b)
```

1 blue  
2 green  
3 black

In [88]:

```
#65
def test(lst):
    result = sorted(lst, key=lambda x: not x)
    return result
nums = [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]
print("\nOriginal list:")
print(nums)
print("\nMove all zero digits to end of the said list of numbers:")
print(test(nums))
```

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]

In [89]:

```
#66
num = [[1,2,3], [4,5,6], [10,11,12], [7,8,9]]
print(max(num, key=sum))
```

[10, 11, 12]

In [90]:

```
#67
list1 = [220, 330, 500]
list2 = [12, 17, 21]
print(all(x >= 200 for x in list1))
print(all(x >= 25 for x in list2))
```

True  
False

In [91]:

```
#68
x = [10, 20, 30]
y = [40, 50, 60]
x[:0] = y
print(x)
```

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

In [92]:

```
#69
import itertools
num = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
print("Original List", num)
num.sort()
new_num = list(num for num, _ in itertools.groupby(num))
print("New List", new_num)
```

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

In [93]:

```
#70
def dict_depth(d):
    if isinstance(d, dict):
        return 1 + (max(map(dict_depth, d.values()))) if d else 0
    return 0
dic = {'a':1, 'b': {'c': {'d': {}}}}
print(dict_depth(dic))
```

```
4
```

In [94]:

```
#71
my_list = [{}, {}, {}]
my_list1 = [{1:2}, {}, {}]
print(all(not d for d in my_list))
print(all(not d for d in my_list1))
```

```
True
False
```