

**Рекурсия:**  
**Данные входные**  
**Список списков**  
**Files = [[1,2,3],[a,b,c],[one,two,three]]**

**Параметры**  
**(откуда они взялись - их поставил**  
**я так я вижу решение)**  
**1 - сам список списков - files**  
**2 - номер текущего вложенного**  
**списка - n**  
**3 - список комбинаций -**  
**заполняем в ходе вызовов**

```
def combine(files, n, pres):  
    flag = n == len(files)-1 # проверяем не последний ли список  
    for x in files[n]:  
        res = copy.copy(pres) # pres дает комбинацию предыдущих списков  
        res.append(x) # в цикле добавляем туда текущий элемент текущего списка  
        if( flag ):  
            print(res) # если список последний комбинация готова - выводим ее на экран  
        else:  
            combine(files,n+1,res) # список не последний нужно добавить из следующего
```

Вызываем рекурсию извне - передаем (files,0,[]) большой список, n = 0 - начинаем с первого списка и пустой список - для заполнения комбинаций - почему??? Потому что в рекурсии внутри его определить нельзя - там он будет создаваться на каждом шаге и мы потеряем предыдущие значения(!!)

*res = copy.copy(pres)*

[ 1, 2, 3 ] - n = 0, [] - обходим циклом список

(n=1, [1]) (n=1, [2]) (n=1, [3]) на каждом шаге цикла мы создаем (pres.copy()) новый список для комбинации и добавляем текущее значение - и вызовем опять эту функцию с нашей текущей комбинацией и номером следующего списка

[ a, b, c ] - будет вызван три раза (files, n=1, [1]) (files, n=1, [2]) (files, n=1, [3]) - n не последний так что повторяем логику первого пункта пример [2]

(n=2, [2, a]) (n=2, [2, b]) (n=2, [2, c]) на каждом шаге цикла мы создаем (pres.copy()) новый список для комбинации и добавляем текущее значение - и вызовем опять эту функцию с нашей текущей комбинацией и номером следующего списка

[ one, two, three ] - последний список тут мы добавляем элемент в комбинацию и выводим на экран для (n=2, [2, b]) проследим вывод

[ 2, b, one ], [ 2, b, two ], [ 2, b, three ]