"1) Создать список из чисел 31, 24, 17 тремя способами: перечисление, на основе другого списка (по сути, создать копию списка), с помощью функции range. Вывести списки на печать. Создать и вывести список, состоящий из полученных списков."

]

"2.1) Создать список из всех симоволов имеющихся во введенной пользователем строке."

]

"2.2) Создать список из всех слов имеющихся во введенной пользователем строке."

]

"2.3) Создать и вывести список из всех цифр, встречающихся в заданной строке."

]

"3) Если в заданный список входит слово \"repeat\", а последний элемент - число, то заменить список на столько же копий всех элементов, не считая последних двух (последние два просто добавить). Например, список ['input', 'string', 'repeat', 3] должен быть заменен на ['input', 'string', 'input', 'string', 'input', 'string', 'repeat', 3]."

]

"4) Создать список из двух произвольных строк s1 и s2, их длин и результата проверки соблюдения лексикографического порядка (идет ли первая строка раньше второй). В зависимости от параметра output путем обращения к элементам списка выводить либо длины строк (output='lengths', в формате \"Длины строк: 4 и 5\"), либо описание порядка (output='order', в формате \"Строка 'первая' идет ПОСЛЕ строки 'вторая'\")."

]

"5.1) Запросить у пользователя 2 целых числа и сохранить их в переменных max\_val, repeat. Создать список из целых чисел со значениями от 1 до max\_val включительно повторяющийся repeat раз.\n",

"5.2) На основе списка из 5.1 созданного при max\_val = 10 создать его копию (значение repeat заранее неизвестно). В копии с помощью среза удалить 80% элементов из середины. Все элементы исходного списка, которые не сохранены в скопированном списке, умножить на десять."

]

"6) Из произвольной строки создать список. Вместо каждой буквы \"s\" (без учета регистра), стоящей не на первом и не на последнем месте, вставить в список строку из двойного предыдущего символа и одного следующего. Например, из строки \"test\\\_Stringss\" должен получиться список ['t', 'e', 'eet', 't', '\\\_', '\\\_\_t', 't', 'r', 'i', 'n', 'g', 'ggs', 's']."

]

"7)\* Организовать заполнение списка заданной длины пользователем через input(). Если на вход подается число (целое или через точку), то записывать в список целое число с округлением вверх. Если \"True\" или \"False\", то как boolean. Иначе - как строку."

]

"8) Заданы две заранее неизвестные строки в которых слова разделены пробелами. Из двух строк составить одну, в котрой слова из первой и второй строки будут чередоваться при этом порядок слов будет сохранен. Для преобразований использовать списки слов.\n",

"\n",

"Пример: 'один два три', 'альфа бетта гамма' -> 'один альфа два бетта три гамма'"

]

"9) Решить здачу 8 используя функцию zip."

]

"10) Выполнить циклический сдвиг слов в произвольной строке (слова разделены пробелами) на заднное пользователем число слов. Решить с помощью списков слов.\n",

"\n",

"Пример \"один два три четыре пять шесть семь\", 3 -> \"четыре пять шесть семь один два три\""

]

"11) В списке целых чисел из диапозона от 1 до 20 заменить все числа не являющиеся простыми на составляющие их простые множители (которые включаются в виде отдельных элементов списка на место исключаемого числа). Каратные простые множетели добавлять количество раз составляющее их кратность в исходном числе.\n",

"\n",

"Пример: [3, 6, 7, 12] -> [3, 2, 3, 7, 2, 2, 3]"

]

" 12) В списке, состоящем из чисел найти значение меньше или равно которому только одно значения из списка."

]

"13) Найти индекс в списке для значения, найденного в задаче 12."

]

"14) Найти индекс второго с конца вхождения значения 1 в список из целых чисел.ь"

]

"15) Реализовать функцию умножения двух матриц, записанных как список списков. Проверить на умножении матрицы на единичную матрицу."

]

"16) Создать копию матрицы, записанной в виде вложенных списков, так что при изменении элементов исходной матрицы, скопированная матрица меняться не будет. "

]

"17) Осортировать слова в строке (с пробелом в качестве разделителя) в лексикографическом порядке.\n",

"\n",

"Пример: 'дыня абрикос яблоко арбуз' -> 'абрикос арбуз дыня яблоко'"

]

"18) Имеется переменная lsls содержащая список списков. Заранее неизвестно, сколько в переменной списков и какой они длины (их длины в общем случае различны). Из этих списков составить список, содержащий поорчедно элементы каждого из вложенных списков, при этом, порядок этих элементов сохраняется. Процедура создания списка останавливается когда хотя бы в одном из исходных вложенных списков будут использованные все элементы. Кроме составленного списка получить список из всех элементов не вошедших в составленный список.\n",

"\n",

"Пример: lsls=[[1,2,3],['a', 'b'], [30, 40, 50, 60]] -> [1, 'a', 30, 2, 'b', 40], [3, 50, 60]"

]