

Python. Списки

октябрь, 2024

Списки в Python — это упорядоченные коллекции элементов, которые могут содержать данные различных типов. Они позволяют хранить и обрабатывать большие объёмы информации.

Каждый элемент — это ссылка на данные об этом элементе в памяти компьютера. Поэтому элементы в списках Python могут быть любого размера и типа — число, строка или ещё один список.

Списки создаются с помощью квадратных скобок, `[]` а элементы внутри списка разделяются запятыми. Например: `my_list = [1, 2, 3, 4, 5]`.

Теорию для данной лабораторной вы можете найти в учебном пособии

Аудиторный практикум

Задание 1

Однажды трое лучших друзей Петя, Вася и Тоня решили создать команду и участвовать в соревновании по программированию. Обычно на соревнованиях по программированию участникам предлагается несколько задач. Еще перед началом дружья договорились, что они будут реализовывать задачу, если как минимум двое из них придумали и уверены в решении, в противном случае ребята не будут писать решение этой задачи.

На этом соревновании организаторы предложили n задач. Для каждой задачи известно, кто из друзей придумал и уверен в ее решении. Помогите друзьям найти количество задач, по которым они будут писать решение.

Входные данные В первой строке входных данных записано единственное целое число n ($1 \leq n \leq 1000$) — количество задач на соревновании. Далее в n строках записано по три целых числа, каждое из которых равно 0 или 1. Если первое число в строке равно 1, то Петя уверен в решении этой задачи, в противном случае он не уверен в решении. Второе число обозначает мнение Васи, а третье — мнение Тони. Числа в строках разделены пробелами.

Выходные данные Выведите единственное целое число — количество задач, которые друзья будут реализовывать на соревновании.

Входные данные	Выходные данные
3 1 1 0 1 1 1 1 0 0	2
2 1 0 0 0 1 1	1

Задание 2

Дан список `[[1, 2, ['Ok!'], 3], ['list', 4], 5]`. Выведите на экран строку `'Ok!'`, используя синтаксис доступа к элементу списка по его индексу.

Задание 3

В данном упражнении вам предстоит разработать программу, в которой у пользователя будет запрошен список слов, пока он не оставит строку ввода пустой. После этого на экране должны быть показаны слова, введенные пользователем, но без повторов, – каждое по одному разу. При этом слова должны быть отображены в том же порядке, в каком их вводили с клавиатуры.

Входные данные	Выходные данные
first second first third second	first second third

Лабораторная работа 8

В данной лабораторной работе присутствуют задания повышенной сложности, поэтому оценивание выполненной лабораторной работы будет происходить следующим образом:

Критерии оценивания лабораторной:

- Чтобы получить оценку 4 (75%), необходимо сделать 7 заданий.
- Чтобы получить оценку 5 (90%), необходимо сделать 8 заданий.
- Чтобы получить оценку 5 (100%), необходимо сделать 9 заданий.

Задание 1

Вы работаете в компании, которая занимается обработкой данных. Вам поступил список данных, в котором перемешаны буквы и цифры. Разбейте этот список на два новых списка: один с буквами, а второй с цифрами. После этого удалите исходный список и выведите на экран два новых списка, каждый на отдельной строке.

Входные данные	Выходные данные
1 a 3 4 b 6	1 3 4 6 a b

Задание 2

Для выигрыша главного приза необходимо, чтобы шесть номеров на лотерейном билете совпали с шестью числами, выпавшими случайным образом в диапазоне от 1 до 49 во время очередного тиража. Напишите программу, которая будет случайным образом генерировать билет.

Входные данные	Выходные данные
	1 45 34 23 31 45

Задание 3

Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.

Входные данные	Выходные данные
1 5 2 4 3	5 4

Задание 4

Напишите программу, которая будет запрашивать у пользователя числа, пока он не пропустит ввод. Сначала на экран должно быть выведено среднее значение введенного ряда чисел, после этого друг за другом необходимо вывести список чисел ниже среднего, равных ему (если такие найдутся) и выше среднего. Каждый список должен предваряться соответствующим заголовком.

Входные данные	Выходные данные
1 3 5 7 2 8	4.333333333333333 1 3 2 5 7 8

Задание 5

Андрей перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать. Программа получает на вход невозрастающую последовательность натуральных чисел, означающих рост каждого человека в строю. После этого вводится число X —

рост Андрея. Все числа во входных данных натуральные и не превышают 230. Выведите номер, под которым Андрей должен встать в строй. Если в строю есть люди с одинаковым ростом, таким же, как у Андрея, то он должен встать после них.

Входные данные	Выходные данные
176 215 207 196 176 168 166	5
176 215 210 207	4

Задание 6

Какое минимальное количество раз вы должны подбросить монетку, чтобы три раза подряд выпал либо орел, либо решка? А какое максимальное количество попыток может для этого понадобиться? А в среднем? В данном упражнении мы выясним это, создав симулятор подбрасывания виртуальной монетки. Напишите программу, использующую для подброса монетки генератор случайных чисел Python. Монетка при этом должна быть правильной формы, что означает равную вероятность выпадения орла и решки. Подбрасывать монетку необходимо до тех пор, пока три раза подряд не выпадет одно значение, вне зависимости от того, орел это будет или решка. Выводите на экран букву О всякий раз, когда выпадает орел, и Р – когда выпадает решка.

Входные данные	Выходные данные
	О Р Р Р (попыток: 4)
	Р О Р О О О (попыток: 6)

Задание 7

Алгоритм Луна (англ. Luhn algorithm), также известный как алгоритм «модуля 10» или «mod 10» — алгоритм вычисления контрольной цифры номера пластиковой карты в соответствии со стандартом ISO/IEC 7812. Назван в честь его создателя, ученого из IBM Ханса Питера Луна. Алгоритм описан в 1954 году, патент получен в 1960 году.

Сам алгоритм не является криптографическим средством, а предназначен в первую очередь для выявления ошибок, вызванных непреднамеренным искажением данных (например, при ручном вводе номера карты, при приёме данных о номере социального страхования по телефону). Позволяет лишь с некоторой степенью достоверности судить об отсутствии ошибок в блоке цифр, но не даёт возможности нахождения и исправления обнаруженной неточности.

Напишите программу для проверки номера банковской карты по алгоритму Луна (необходимо просуммировать все четные по номеру цифры последовательности, далее прибавить к сумме все нечетные по номеру цифры, помноженные на 2, при этом если произведение получается больше 9, то из него вычитается 9. Сумма должна быть кратной 10).

Входные данные	Выходные данные
4276440013361511	Корректный номер
4276440013361512	Некорректный номер
42761336512	Некорректный номер

Задание 8

Иногда некоторые слова вроде «localization» или «internationalization» настолько длинны, что их весьма утомительно писать много раз в каком либо тексте.

Будем считать слово слишком длинным, если его длина строго больше 10 символов. Все слишком длинные слова можно заменить специальной аббревиатурой.

Эта аббревиатура строится следующим образом: записывается первая и последняя буква слова, а между ними — количество букв между первой и последней буквой (в десятичной системе счисления и без ведущих нулей).

Таким образом, «localization» запишется как «l10n», а «internationalization» как «i18n».

Вам предлагается автоматизировать процесс замены слов на аббревиатуры. При этом все слишком длинные слова должны быть заменены аббревиатурой, а слова, не являющиеся слишком длинными, должны остаться без изменений.

Входные данные В первой строке содержится целое число n ($1 \leq n \leq 100$). В каждой из последующих n строк содержится по одному слову. Все слова состоят из малых латинских букв и имеют длину от 1 до 100 символов.

Выходные данные Выведите n строк. В i строке должен находиться результат замены i -го слова из входных данных.

Входные данные	Выходные данные
3	word
word	l10n
localization	i18n
internationalization	

Задание 9

Недавно Юра поступил в университет. У Юры есть друг Леша, который поступил вместе с ним, и теперь они заселяются в общежитие.

Юра и Леша хотят жить в одной комнате. Всего в общежитии есть n комнат. В данный момент в комнате с номером i живут p_i человек, когда всего в этой комнате может жить q_i человек ($p_i \leq q_i$). Посчитайте, сколько комнат общежития смогут вместить Юру и Лешу вместе?

Входные данные В первой строке содержится единственное целое число n ($1 \leq n \leq 100$) — количество комнат.

В i -й из n последующих строк содержатся два целых числа p_i и q_i ($0 \leq p_i \leq q_i \leq 100$) — количество людей, которые уже живут в комнате, и максимальное допустимое количество людей, живущих в i -й комнате.

Выходные данные Выведите одно целое число — количество комнат, в которые Юра с Лешей могут заселиться.

Входные данные	Выходные данные
3	0
1 1	
2 2	
3 3	
3	2
1 10	
0 10	
10 10	

Решения аудиторного практикума

Задание 1

```
while 1:
    try:
        n = int(input("Введите количество задач:"))
        if n > 0:
            break
    except ValueError:
        print("Ошибка!")

count = 0

for i in range(n):
    s = input(">").split()

    voices = 0

    for j in s:
        if j != '1' and j != '0':
            print("Ошибка! Не корректное значение. Программа завершает свою работу")
            exit()
        else:
            voices += int(j)

    if voices >= 2:
        count += 1

print("Всего будет решено:", count)
```

Задание 2

```
a = [[1, 2, ['0k!', 3]], ['list', 4], 5]

part1 = a[0]
part2 = part1[2]
part3 = part2[0]
print(part3)
```

Задание 3

```
words = []
while 1:
    in_word = input()
    if in_word != "":
        words.append(in_word)
    else:
        break
```

```
res = []  
for word in words:  
    if word not in res:  
        res.append(word)
```

```
print(res)
```

или красивый вариант вывода:

```
for word in res:  
    print(word)
```