списке

Список представляет собой последовательность элементов, пронумерованных от 0, как символы в строке.

Primes = [2, 3, 5, 7, 11, 13]

Rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue']

В списке Primes — 6 элементов, а именно:

Primes[0] = 2, Primes[1] = 3, Primes[2] = 5, Primes[3] = 7,

Primes[4] = 11, Primes[5] = 13.

Список Rainbow состоит из 5 элементов, каждый из которых является строкой.

Также как и символы в строке, элементы списка можно индексировать отрицательными числами с конца, например,

Primes[-1] = 13, Primes[-6] = 2.

- Длину списка, то есть количество элементов в нем, можно узнать при помощи функции len, например, len(Primes) =6.

В отличие от строк, элементы списка можно изменять, присваивая им новые значения.

```
Rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue']

Rainbow[0] = 'красный‘

for i in range(len(Rainbow)):
    print(Rainbow[i])
```

Пример создания списка

```
a = []  # заводим пустой список
n = int(input())  # считываем количество элемент в списке
for i in range(n):
    new_element = int(input())
    a.append(new_element)  # добавляем его в список
```

a.append(int(input()))

Пример создания списков

```
a = [1, 2, 3]
b = [4, 5]
c = a + b
d = b * 3
print([7, 8] + [9])
print([0, 1] * 3)
```

Пример создания списка

```
a = [0] * int(input())
for i in range(len(a)):
    a[i] = int(input())
```

```
a = [1, 2, 3, 4, 5]
for elem in a:
    print(elem, end=' ')
```

В этом примере элементы списка выводятся в одну строку, разделенные пробелом, при этом в цикле меняется не индекс элемента списка, а само значение переменной

в цикле переменная elem будет последовательно принимать значения 'red', 'green', 'blue'.

Очень важная часть идеологии Питона — это цикл for, который предоставляет удобный способ перебрать все элементы некоторой последовательности.

В этом отличие Питона от Паскаля, где вам обязательно надо перебирать именно индексы элементов, а не сами элементы.

Последовательностями в Питоне являются строки, списки, значения функции range() (это не списки), и ещё кое-какие другие объекты.

из строки надо выбрать все цифры и сложить их в массив как числа.

```
s = 'ab12c59p7dq‘
digits = []
for symbol in s:
    if '1234567890'.find(symbol) != -1:
        digits.append(int(symbol))
print(digits)
```

Метод split

```
s = input()  # s ='1 2 3‘
a = s.split()  # a = ['1', '2', '3']
```

```
a = input().split()
for i in range(len(a)):
    a[i] = int(a[i])
a = [1, 2, 3]
```

```
a = '192.168.0.1'.split('.')
```

Метод join

```
a = ['red', 'green', 'blue']
print(' '.join(a))
# вернёт red green blue
print(''.join(a))
# вернёт redgreenblue
print('***'.join(a))
# вернёт red***green***blue
```

```
a = [1, 2, 3]
print(' '.join([str(i) for i in a]))
```

Генераторы списков

- Для создания списка, заполненного одинаковыми элементами

```
n = 5
a = [0] * n
```

- получить список, заполненный случайными числами от 1 до 9

```
from random import randrange
n = 10
a = [randrange(1, 10) for i in range(n)]
```

- Создать список, состоящий из 0

```
a = [0 for i in range(5)]
```

- Создать список, заполненный квадратами целых чисел

```
n = 5
a = [i ** 2 for i in range(n)]
```

Срезы

- Со списками, так же как и со строками, можно делать срезы. А именно:

- A[i:j] срез из j-i элементов A[i], A[i+1], ..., A[j-1].

- A[i:j:-1] срез из i-j элементов A[i], A[i-1], ..., A[j+1] (то есть меняется порядок элементов).

- A[i:j:k] срез с шагом k: A[i], A[i+k], A[i+2*k],... . Если значение k<0, то элементы идут в противоположном порядке.

Списки, в отличии от строк, являются изменяемыми объектами: можно отдельному элементу списка присвоить новое значение. Но можно менять и целиком срезы.

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

A[2:4] = [7, 8, 9]

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

A[::-2] = [10, 20, 30, 40]

Здесь A[::-2] — это список из элементов A[-1], A[-3], A[-5], A[-7], которым присваиваются значения 10, 20, 30, 40 соответственно.

Обратите внимание, A[i] — это элемент списка, а не срез!

Операции со списками

Со списками можно легко делать много разных операций.

x in A Проверить, содержится ли элемент в списке. Возвращает True или False

x not in A То же самое, что not(x in A)

min(A) Наименьший элемент списка

max(A) Наибольший элемент списка

A.index(x) Индекс первого вхождения элемента x в список, при его отсутствии генерирует исключение ValueError

A.count(x) Количество вхождений элемента x в список

Задачи для решения:

1. Выведите все элементы списка с четными индексами (то есть A[0], A[2], A[4], ...).
2. Выведите все четные элементы списка.

3. Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.
4. Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет — не выводите ничего. Если таких пар соседей несколько — выведите первую пару.

1. Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда не учитываются, поскольку у них недостаточно соседей.
2. Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.
3. Петя 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 Пети. Все числа во входных данных натуральные и не превышают 200.

Выведите номер, под которым Петя должен встать в строй. Если в строю есть люди с одинаковым ростом, таким же, как у Пети, то он должен встать после них.