

Программирование на языке Python

Массивы (списки)

Что такое массив?



Как ввести 10000 переменных?

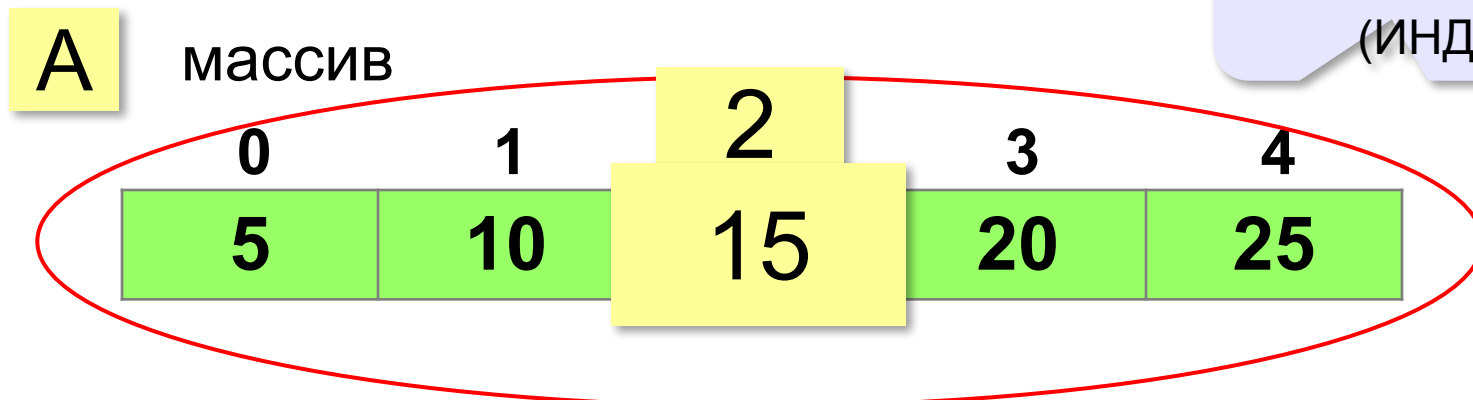
Массив – это группа переменных одного типа, расположенных в памяти рядом (в соседних ячейках) и имеющих общее имя. Каждая ячейка в массиве имеет уникальный номер (индекс).

Надо:

- выделять память
- записывать данные в нужную ячейку
- читать данные из ячейки

Что такое массив?

! Массив = таблица!



A[0] **A[1]** **ЗНАЧЕНИЕ**
элемента массива **A[4]**

A[2]

НОМЕР (ИНДЕКС)
элемента массива: 2

ЗНАЧЕНИЕ
элемента массива: 15

Алгоритм решения задач с использованием массива

1. Создание списка
2. Ввод элементов (формирование) списка
3. Вывод элементов списка (если используем random)
4. Обработка элементов списка (решение исходной задачи)
5. Вывод результата / вывод нового списка

Массивы в Python: списки

```
A = [1, 3, 4, 23, 5]
```

```
A = [1, 3] + [4, 23] + [5]
```

```
[1, 3, 4, 23, 5]
```

```
A = [0] * 10
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```



Что будет?

Создание массива из N элементов:

```
N = 10
```

```
A = [0] * N
```

```
N = int(input())
```

```
A = [0] * N
```

1 способ Заполнение массива с клавиатуры

Ввод с клавиатуры по одному элементу в строке:

```
N = int(input()) # размер массива
A = [0]*N        # выделить память
for i in range(N):
    A[i] = int(input())
```

```
5
-4
6
-8
7
11
```

2 способ Заполнение случайными числами

`random.uniform(a, b)`

псевдослучайное вещественное
число в диапазоне от a до b

`random.randint(a, b)`

псевдослучайное целое число в
диапазоне от a до b

Заполнение случайными числами

из библиотеки (модуля)
random

ВЗЯТЬ функцию randint

```
from random import randint
N = int(input())
A = [0]*N
for i in range(N):
    A[i] = randint(20, 100)
```

```
from random import uniform
N = int(input())
A = [0]*N
for i in range(N):
    A[i] = uniform(0.5, 15.5)
```


Заполнение случайными числами

подключение
библиотеки (модуля)
random

```
from random import *
N = int(input())
A = [0]*N
for i in range(N):
    A[i] = randint(-20,100)
```

```
from random import *
N = int(input())
A = [0]*N
for i in range(N):
    A[i] = uniform(0.5,15.5)
```

~~3 способ Заполнение при помощи формулы~~

```
N = int(input())  
A = [0]*N  
for i in range(N):  
    A[i] = 3*i+i
```

Вывод массива на экран

В строчку через пробел:

```
for i in range(N):  
    print ( A[i], end = " " )
```

1 2 3 4 5

пробел после
вывода
очередного числа

или так:

```
print ( *A ) ↔ print ( 1 , 2 , 3 , 4 , 5 )
```

разбить список
на элементы

- 1. Ввести элементы массива с клавиатуры. Вывести их в строку.**
- 2. Ввести элементы массива при помощи датчика случайных чисел в диапазоне [0, 60]. Вывести в столбец.**
- 3. Заполнить элементы массива в диапазоне [-21.5, 52.5]. Вывести в столбец.**
- 4. Заполнить массив нечетными числами и вывести в строку.**
- 5. Ввести элементы массива при помощи формулы $i*i+5$. Вывести в строку.**

Ввод элементов массива в строку через пробел

С клавиатуры:

```
A = list( map(int, input().split()) )
```

построить
СПИСОК

применить `int` ко
всем элементам `A`

Ввод элементов массива в строку через пробел

При помощи датчика случайных чисел:

```
from random import *  
A = [ randint(20, 100)  
      for x in range(10) ]
```

случайные
числа

Вывод массива на экран

В строчку через пробел:

```
for i in range(N):
```

```
    print ( A[i], end= " " )
```

1 2 3 4 5

пробел после
вывода
очередного числа

или так:

```
print ( *A )
```



```
print (1, 2, 3, 4, 5)
```

разбить список
на элементы

Ввод массива

В строчку через пробел:

```
a = input().split()
for i in range(len(a)):
    if int(a[i])%2==0:
        print(a[i])
```

- строка целиком считывается **функцией input()** – *данные не числа, а строки*
- **метод строки split()**, возвращающий список строк, которые получатся, если исходную строку разрезать на части по пробелам
- **функции len** определяет длину списка, то есть количество элементов в нем

Ввод массива с клавиатуры

Ввод всех чисел в одной строке:

```
a = input()      # "1 2 3 4 5"
s = a.split()    # ["1", "2", "3", "4", "5"]
A = [ int(x) for x in s ]
                # [1, 2, 3, 4, 5]
```

или так:

```
s = input().split() # ["1", "2", "3", "4", "5"]
A = list( map(int, s) ) # [1, 2, 3, 4, 5]
```

построить
список

применить `int` ко
всем элементам `s`

Программирование на языке Python

Основные типы задач

Сумма элементов списка

Задача. Найти сумму чётных элементов массива.

```
N = int(input())
A = [0]*N
for i in range(N):
    A[i] = int(input())

s = 0
for i in range(N):
    if A[i] % 2 == 0:
        s += A[i]
print(s)
```

Количество элементов списка

Задача. Найти количество элементов списка, кратных 5 и стоящих на чётных местах.

```
N = int(input())
A = [0]*N
from random import *
for i in range(N):
    A[i] = randint(-55,10)

k = 0
for i in range(N):
    if A[i] % 5 == 0 and i % 2 == 0:
        k += 1
print ( k )
```

Среднее арифметическое

Задача. Найти среднее арифметическое элементов массива, которые оканчиваются на цифру 5.

```
count = 0
summa = 0
for x in A:
    if x % 10 == 5:
        count += 1
        summa += x
print ( summa/count )
```



Как определить, что оканчивается на 5?

среднее
арифметическое

или так:

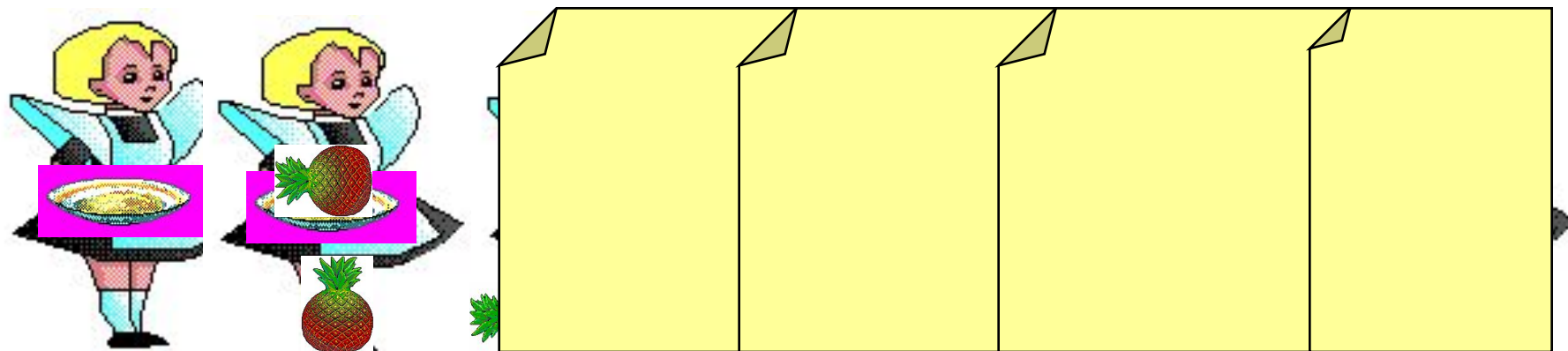
```
B = [ x for x in A
      if x % 10 == 5 ]
print ( sum(B)/len(B) )
```

отбираем нужные

Максимальный элемент

Задача: найти в массиве максимальный элемент.

Алгоритм:



Решение:

- 1) считаем, что первый элемент – максимальный
- 2) просмотреть остальные элементы массива:
если очередной элемент $> M$,
то записать $A[i]$ в M
- 3) вывести значение M

Максимальный элемент

```
M = A[0]
for i in range(1, N):
    if A[i] > M:
        M = A[i]
print ( M )
```



Если `range(N)` ?

Варианты в стиле Python:

```
M = A[0]
for x in A:
    if x > M:
        M = x
```



Как найти его номер?

```
M = max ( A )
```

Максимальный элемент и его номер

```
M = A[0]; nMax = 0
for i in range(1, N):
    if A[i] > M:
        M = A[i]
        nMax = i
print( "A[" , nMax, "]=" , M, sep = "" )
```



Что можно улучшить?



По номеру элемента можно найти значение!

```
nMax = 0
for i in range(1, N):
    if A[i] > A[nMax]:
        nMax = i
print( "A[" , nMax, "]=" , A[nMax] , sep = "" )
```


Максимальный элемент и его номер

Вариант в стиле Python:

```
M = max (A)
nMax = A . index (M)
print ( "A[" , nMax , "]" = " , M , sep = " " )
```

номер заданного
элемента (первого из...)

Задачи (без функций **min** и **max**)

«3»: Заполнить массив из 10 элементов случайными числами в интервале [10,100] и найти минимальный и максимальный элементы массива и их номера.

Пример:

Массив :

39 52 84 77 45 32 19 38 49 85

Минимальный элемент: A[6]=19

Максимальный элемент: A[9]=85

Задачи (без функций **min** и **max**)

«4»: Заполнить массив из 10 элементов случайными числами в интервале [10,100] и найти минимальный и максимальный элементы из **чётных** элементов массива.

Пример:

Массив :

39 52 84 77 45 32 19 38 49 85

Минимальный чётный: 32

Максимальный чётный: 84

Задачи (без функции **max**)

«5»: Ввести с клавиатуры массив из 5 элементов и найти два максимальных элемента массива и их номера.

Пример:

Массив :

5 5 3 4 1

Максимальный элемент: $A[1]=5$

Второй максимум: $A[2]=5$

Задачи

«6»: Введите массив с клавиатуры и найдите (за один проход) количество элементов, имеющих максимальное значение.

Пример:

Массив :

3 4 5 5 3 4 5

Максимальное значение 5

Количество элементов 3