ПРОГРАММИРОВАНИЕ

МАССИВЫ

(СПИСКИ)

ЧТО ТАКОЕ МАССИВ?

? Как ввести 10000 переменных?

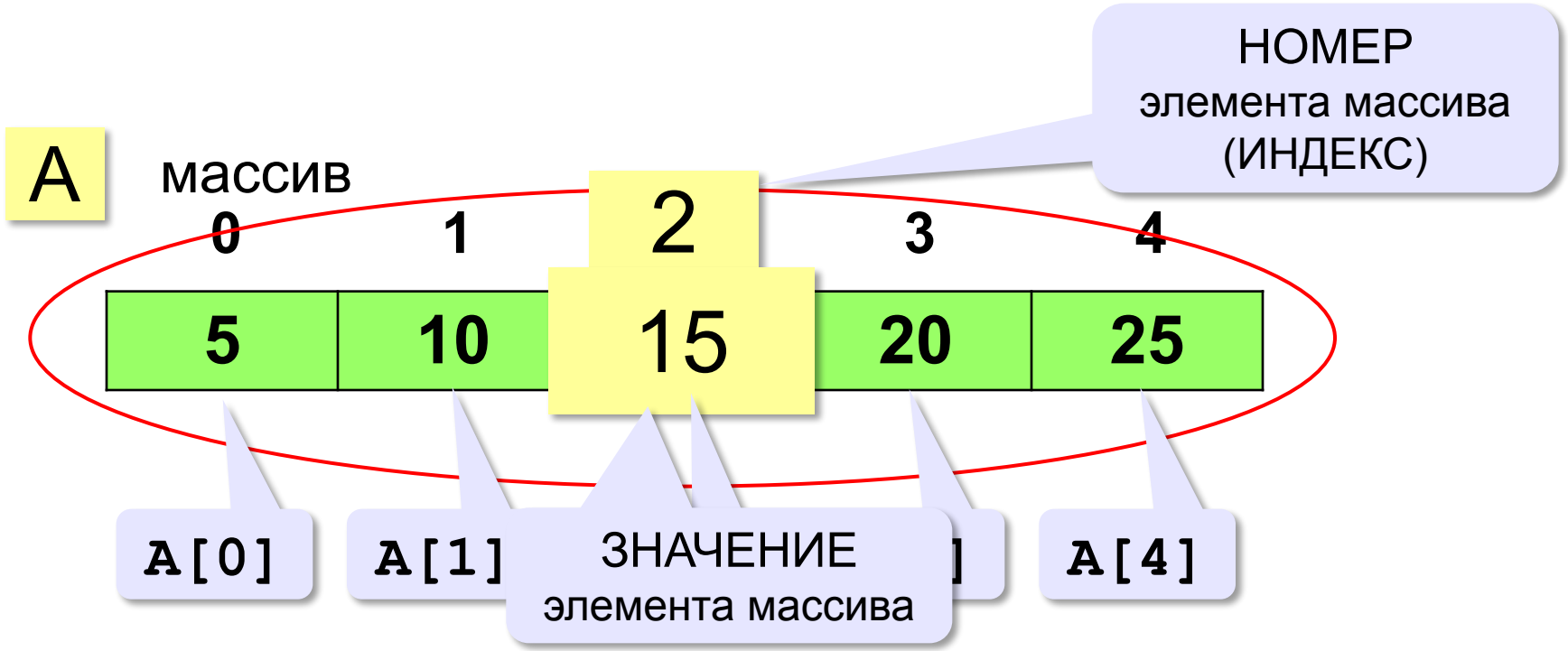
Массив – это группа переменных одного типа, расположенных в памяти рядом (в соседних ячейках) и имеющих общее имя.

Надо:

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

```
Primes = [2, 3, 5, 7, 11, 13]
Rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']
```

ОБРАЩЕНИЕ К ЭЛЕМЕНТУ МАССИВА



Индекс элемента — это значение, которое указывает на конкретный элемент массива.

Нумерация с нуля!

ОБРАЩЕНИЕ К ЭЛЕМЕНТУ МАССИВА

A[2]

ИНДЕКС элемента массива: 2

ЗНАЧЕНИЕ элемента массива

0	1	2	3	4
23	12	7	43	51

```
i = 1
A[2] = A[i] + 2*A[i-1] + A[2*i+1]
print( A[2]+A[4] )
```

? Что получится?

```
A[2] = A[1] + 2*A[0] + A[3]
print( A[2]+A[4] )
```

```
101
152
```

СОЗДАНИЕ МАССИВА

```
A = [11, 22, 35, 41, 53]
```

11	22	35	41	53

```
A = [11, 22] + [35, 41] + [53]
```

```
A = [11]*5
```

11	11	11	11	11

ЧТО НЕВЕРНО?

```
A = [1, 2, 3, 4, 5]
x = 1
print( A[x-8] )
A[x+4] = A[x-1] + A[2*x]
```

? Что плохо?

```
print( A[-7] )
A[5] = A[0] + A[2]
```

Выход за границы массива — это обращение к элементу с индексом, который не существует в массиве.

ПЕРЕБОР ЭЛЕМЕНТОВ МАССИВА

```
N = 10
A = [0]*N   # память уже выделена
```

Перебор элементов: просматриваем все элементы массива и, если нужно, выполняем с каждым из них некоторую операцию.

```
for i in range(N):
   # здесь работаем с A[i]
```

ЗАПОЛНЕНИЕ МАССИВА

[0, 2, 3, …, N-1]

```
for i in range(N):
  A[i] = i
```

? Что произойдёт?

В развёрнутом виде

```
A[0] = 0
A[1] = 1
A[2] = 2
...
A[N-1] = N-1
```

0	1	2	...	N-1

В стиле Python:

```
A = [ i for i in range(N) ]
```

ЗАПОЛНЕНИЕ МАССИВА В ОБРАТНОМ ПОРЯДКЕ

N	…	3	2	1

```
A[0] = N
A[1] = N-1
A[2] = N-2
...
A[N-1] = 1
```

```
X = N
for i in range(N):
    A[i] = X
    X = X - 1
```

? Как меняется X?

X = N, N-1, …, 2, 1

начальное значение

уменьшение на 1

ЗАПОЛНЕНИЕ МАССИВА В ОБРАТНОМ ПОРЯДКЕ

N	...	3	2	1

`A[i] = X`

? Как связаны `i` и `X`?

i	X
0	N
1	N-1
2	N-2
...	...
N-1	1

+1

–1

```
for i in range(N):
  A[i] = N - i
```

В стиле Python:

```
A = [ N-i
   for i in range(N) ]
```

! Сумма `i` и `X` не меняется!

`i + X = N`

`X = N - i`

ВЫВОД МАССИВА НА ЭКРАН

Весь массив сразу:

```
print( A )
```

```
[1,2,3,4,5]
```

По одному элементу:

```
for i in range(N):
  print( A[i] )
```

в столбик

или так:

```
for x in A:
  print( x )
```

для всех элементов в массиве A

? Как вывести в строчку?

```
for x in A:
  print( x, end=" " )
```

пробел между элементами

ВЫВОД МАССИВА НА ЭКРАН (PYTHON)

```
[1,2,3,4,5]
```

```
print ( *A )
```

⟷

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

разбить список
на элементы

```
1 2 3 4 5
```

ВВОД С КЛАВИАТУРЫ

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

? Что плохо?

или так:

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

С подсказкой для ввода:

```
for i in range(N):
  s = "A[" + str(i) + "]="
  A[i] = int(input(s))
```

```
A[0] = 5
A[1] = 12
A[2] = 34
A[3] = 56
A[4] = 13
```

ВВОД С КЛАВИАТУРЫ (PYTHON)

Ввод всех чисел в одной строке:

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

или так:

```
A = [int(x) for x in input().split()]
```

В ДРУГИХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

Паскаль:

объявление массива

```
const N = 10;
var A: array[0..N-1] of integer;
...
for i:=0 to N-1 do
  A[i] = i;
for i:=0 to N-1 do
  write(A[i], ' ');
```

В ДРУГИХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

C++:

```
int A[N], i;
for (i = 0; i < N; i++)
  A[i] = i;
for (i = 0; i < N; i++)
  cout << A[i] << " ";
```

! Нумерация элементов всегда с нуля!

ЗАПОЛНЕНИЕ СЛУЧАЙНЫМИ ЧИСЛАМИ

из библиотеки (модуля) random

взять функцию randint

```
from random import randint
N = 10        # размер массива
A = [0]*N     # выделить память
for i in range(N):
  A[i] = randint(20,100)
```

В краткой форме:

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

ОБРАБОТКА ЭЛЕМЕНТОВ МАССИВА

```
N = 10
A = [0]*N   # память уже выделена
```

```
for i in range(N):
   # здесь работаем с A[i]
```

Вывести на экран в столбик:

```
for i in range(N):
   print( A[i] )
```

Что вместо «???»?

Вывести на экран в строчку:

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

```
print( *A )
```

ОБРАБОТКА ЭЛЕМЕНТОВ МАССИВА

Вывести числа, на 1 большие, чем `A[i]`:

```
for i in range(N):
  print( A[i]+1 )
```

! Что вместо «???»?

Вывести последние цифры:

```
for i in range(N):
  print( A[i]%10 )
```

ОБРАБОТКА ЭЛЕМЕНТОВ МАССИВА

Заполнить нулями:

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

Увеличить на 1:

```
for i in range(N):
    A[i] += 1
```

Умножить на 2:

```
for i in range(N):
    A[i] *= 2
```

! Что вместо «???»?

ОБРАБОТКА ЭЛЕМЕНТОВ МАССИВА

- Для списков целиком определены операции сложения списков и умножение списка на число.

```
print([7, 8] + [9])
print([0, 1] * 3)
```

```
[7, 8, 9]
[0, 1, 0, 1, 0, 1]
```

ФУНКЦИИ LEN

- Длину списка, то есть количество элементов в нём, можно узнать при помощи функции **len**.

```
len(Primes) == 6
```

Можно создать пустой список (не содержащий элементов длины 0).

a=[ ] – пустой список

МЕТОДА **APPEND**

В конец списка можно добавлять элементы при помощи метода **append**.

```
a = []  # заводим пустой список
n = int(input())  # считываем количество элемент в списке
for i in range(n):
    new_element = int(input())  # считываем очередной элемент
    a.append(new_element)  # добавляем его в список
    # последние две строки можно было заменить одной:
    # a.append(int(input()))
print(a)
```

SPLIT

Элементы списка могут вводиться по одному в строке, в этом случае строку целиком можно считать функцией **input()**. После этого можно использовать метод строки **split(),** возвращающий список строк, которые получаются, если исходную строку разрезать на части по пробелам.

```
# на вход подаётся строка
# 1 2 3
s = input()  # s == '1 2 3'
a = s.split()  # a == ['1', '2', '3']
```

SPLIT

У этого метода есть необязательный параметр, который определяет, какая строка будет использоваться в качестве разделителя между элементами списка.

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

JOIN

Join – метод, позволяющий вывести список. У этого метода один параметр: список. Возвращается строка. Полученная соединением элементов переданного списка в одну строку.

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

РЕШЕНИЕ ЗАДАЧ

1. Напишите программу формирования одномерного массива данных из 10 целочисленных элементов, лежащих в интервале от –10 до 10, и вывода суммы элементов этого массива на экран.

2. Напишите программу формирования одномерного массива данных из 10 целочисленных элементов, лежащих в интервале от 1 до 100, и вывода суммы и количества элементов этого массива кратных 5 и оканчивающихся нулём.

ЗАДАЧИ

1. Напишите программу формирования одномерного массива данных из 10 целочисленных элементов, лежащих в интервале от −10 до 10, и вывода суммы элементов этого массива на экран.

```python
import random
n=10
a=[]
s=0
for i in range (0,n):
  a.append(random.randint(-10, 10))
  s+=a[i]
print(a)
print(s)
```

```pascal
const n=10;
var a: array[1..n] of integer;
    s,i: integer;
begin
  s:=0;
  for i:=1 to n do
    begin
    a[i]:=random(20)-10;
    s:=s+a[i];
    end;
  for i:=1 to n do
    write(a[i], ' ');
  writeln (s);
end.
```

ЗАДАЧИ

2. Напишите программу формирования одномерного массива данных из 10 целочисленных элементов, лежащих в интервале от 1 до 100, и вывода суммы и количества элементов этого массива кратных 5 и оканчивающихся нулём.

```python
import random
n=10
a=[]
s=0
k=0
for i in range (0,n):
  a.append(random.randint(1, 100))
for i in range (0,n):
  if (a[i]%5==0) and (a[i]%10==0):
    s+=a[i]
    k+=1
print(a)
print(s,' ',k)
```

```pascal
const n=10;
var a: array[1..n] of integer;
    k, s, i: integer;
begin
  for i:=1 to n do
    a[i]:=random(101);
  s:=0;
  k:=0;
  for i:=1 to n do

  begin
  if (a[i] mod 5=0) and
     (a[i] mod 10=0) then
    begin
    s:=s+a[i];
    k:=k+1;
    end;
  write(a[i], ' ');
  end;
 writeln;
 write(s, ' ',k);
end.
```

САМОСТОЯТЕЛЬНАЯ РАБОТА

1. ***Выполните тест***: http://kpolyakov.spb.ru/school/test8a/py24x.htm

2. ***Напишите программу, протестируйте на компьютере.***

   2.1. Напишите программу формирования одномерного массива данных из 10 целочисленных элементов, лежащих в интервале от –10 до 10, и вывода этого массива на экран.

   2.2. Напишите программу формирования одномерного массива данных из 10 целочисленных элементов, лежащих в интервале от –10 до 10, и вывода количества положительных (включая ноль) и количества отрицательных элементов этого массива на экран.

   2.3. Напишите программу формирования одномерного массива данных A из 10 целочисленных элементов, лежащих в интервале от –10 до 10, и массива B, который будет заполнен индексами чётных (по значению) элементов массива A.

Задание 1

Python: 24x - Программы с циклами (ОГЭ)

1. Определите, что будет напечатано в результате работы следующей программы.

```
s = 1
for k in range(1, 8):
  s = s + k;
print( s );
```

Ответ: 29

2. Определите, что будет напечатано в результате работы следующей программы.

```
s = 0
for k in range(1, 12):
  s = s + 12
print( s );
```

Ответ: 132

3. Определите, что будет напечатано в результате работы следующей програ

```
s = 0
for k in range(2, 13):
  s = s + 11
print( s );
```

Ответ: 121

4. Определите, что будет напечатано в результате работы следующей програ

```
s = 100
for k in range(11):
  s = s - 5
print( s );
```

Ответ: 45

5. Определите, что будет напечатано в результате работы следующей программы.

```
s = 0
for k in range(14, 19):
  s = s + 7
print( s );
```

Ответ: 35

6. Определите, что будет напечатано в результате работы следующей программы.

```
u = 26
for i in range(1, 6):
  u = u - i
print( u );
```

Ответ: 11

7. Определите, что будет напечатано в результате работы следующей программы.

```
s = 1
for n in range(3, 9):
  s = s * 2
print( s );
```

Ответ: 64

9. Определите, что будет напечатано в результате работы следующей программы.

```
k = 2
for i in range(3):
  k = 3*k + i
print( k );
```

Ответ: 59

10. Определите, что будет напечатано в результате работы следующей программы.

```
s = 3
for i in range(2, 6):
  s = s + 2*i
print( s );
```

Ответ: 31

http://kpolyakov.spb.ru/school/test8a/py24x.htm