

ОДНОМЕРНЫЕ МАССИВЫ

Ты узнаешь:

что такое одномерный массив;

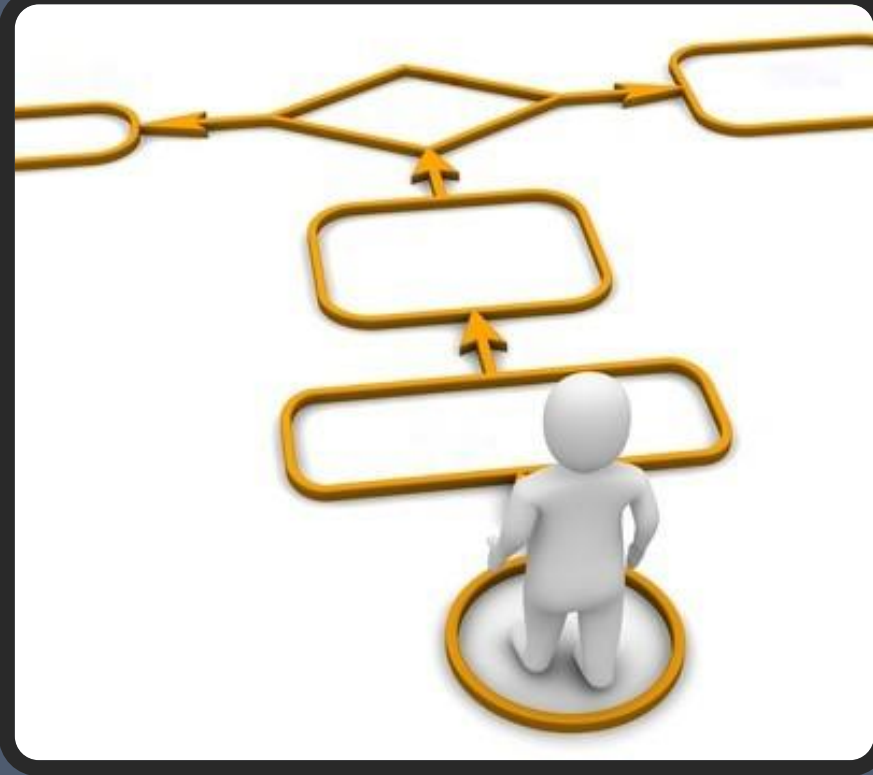
какие бывают операторы для работы с одномерными массивами на языке программирования Python.

Ты научишься:

описывать одномерные массивы на языке программирования Python;

организовывать ввод и вывод элементов одномерного массива;

создавать программы на языке программирования Python с использованием одномерных массивов.



Базовым типом структуры данных в Python является список. В других языках программирования такой структурой является массив.

В Python для работы с массивом подключают модуль `array`, но чаще всего используют список.

Отличие списков от массивов заключается в функциях, которые можно выполнять с ними. Список более широкое понятие. В массиве все элементы имеют одинаковый тип, а в списке

Ключевые понятия

Список – набор элементов произвольного типа.

Массив – набор данных одного типа, записанных под определенным именем.

Длина (размер) массива – количество элементов в массиве.

Индекс – порядковый номер элемента в массиве.

Одномерный массив – массив, в котором для задания элемента требуется один индекс.

Способы создания массивов (списков)



1. Чтобы задать массив (список) в программе, можно указать его имя и в квадратных скобках через запятую перечислить элементы. Значения строковых элементов записываются в апострофах.

Способы создания массивов (списков)



2. Можно описать пустой список и с помощью цикла организовать ввод элементов с клавиатуры.

Способы создания массивов (списков)



3. Заполнение массива по определенному правилу. В данном случае можно использовать генератор списка.

Тип данных массив

Массив — это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип (базовый).

Особенности:

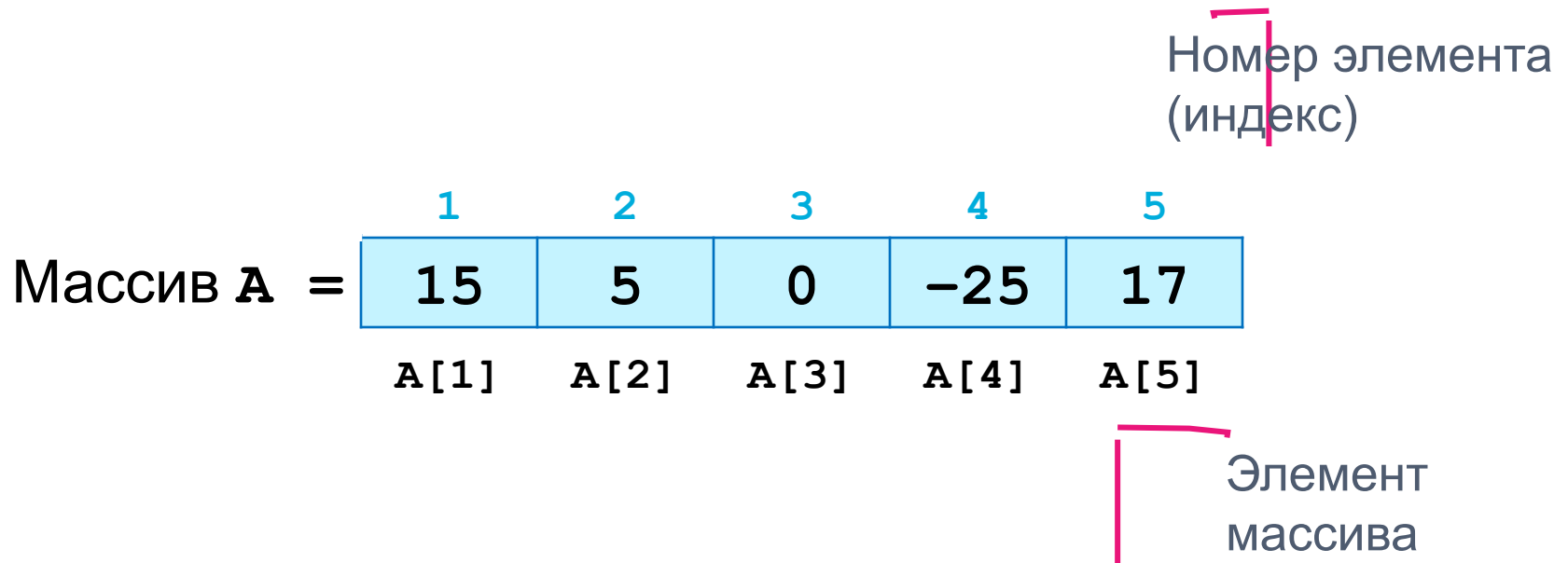
- все элементы имеют *один тип*;
- весь массив имеет *одно имя*;
- элементы упорядочены по *индексам (номерам)*.

Типы массивов:

- одномерный массив;
- двумерный массив;
- ...
- n -мерный массив.



Структура одномерного массива



Элементы массива упорядочены (урегулированы) по *индексам*. Доступ к каждому элементу осуществляется путем *индексирования*, т. е. указания имени массива и номера элемента.

Пример: $a[1] = 15$.

Описание (объявление) массива

Формат блока (в разделе описания типов):

```
Type <имя типа> = array [тип индекса] of <тип элементов>;  
Var <имя массива> : <имя типа>;
```

Формат блока (в разделе описания переменных):

```
Var <имя массива> = array [тип индекса] of <тип  
элементов>;
```

Пример:



```
Const n = 20;  
Type Mas = array [1..n] of Integer;  
Var a, b : Mas;  
      c : array [1..26] of Char;
```



Заполнение массива

Формат блока (ввод элементов с клавиатуры):

```
For <индекс> := 1 to <размер массива> do  
  ReadLn (<имя массива> [<индекс>]);
```

Пример:

```
For i := 1 to n do  
  ReadLn (a[i]);
```

?

a[1] = 15

a[2] = 5

a[3] = 0

a[4] = -25

a[5] = 17

?

?

Назовите главный недостаток заполнения массива путем ввода элементов с помощью клавиатуры.

Вывод массива на экран

Формат блока:

```
For <индекс> := 1 to <размер массива> do  
  Write (<имя массива>[индекс]);
```

Пример:

```
WriteLn ('Массив a:');  
For i := 1 to n do  
  Write (a[i]:4);
```

Результат:

```
Массив a:  
  1  25  -5  10  2
```

Заполнение массива по формуле

Формат блока (ввод элементов по формуле):

```
For <индекс> := 1 to <размер массива> do  
  <имя массива>[<индекс>] := <формула>;
```

Пример:



```
a[1] := 2;  
For i := 2 to n do  
  a[i] := a[i-1] + 2;
```

```
a[1] = 2  
a[2] = 4  
a[3] = 6  
a[4] = 8  
a[5] = 10
```

Задача № 1

Сформировать и вывести на экран массив из 10 первых чисел Фибоначчи.

Заполнение массива с помощью генератора случайных чисел

Функция Random [0;N):

```
x := random;           [0;1)
x := random (50);      [0;49]
x := random (100)+50;  [50;149]
x := random (100)-50;  [-50;49]
x := random (254)-123; [-123;130]
```

Формат блока (использование функции Random):

```
For <индекс> := 1 to <размер массива> do
  <имя массива>[<индекс>] := Random[ (range) ] ;
```

Пример:

```
For i := 1 to n do
  a[i] := random(101);
```

Обработка элементов массива

Вычисление суммы элементов массива

Задача № 2

Ввести с клавиатуры массив из 5 элементов. Найти сумму всех элементов массива.

Входные данные:

	n				
i	1	2	3	4	5
a	15	5	0	-25	10

Выходные данные:

$$\text{sum} = a[1] + a[2] + a[3] + a[4] + a[5]$$

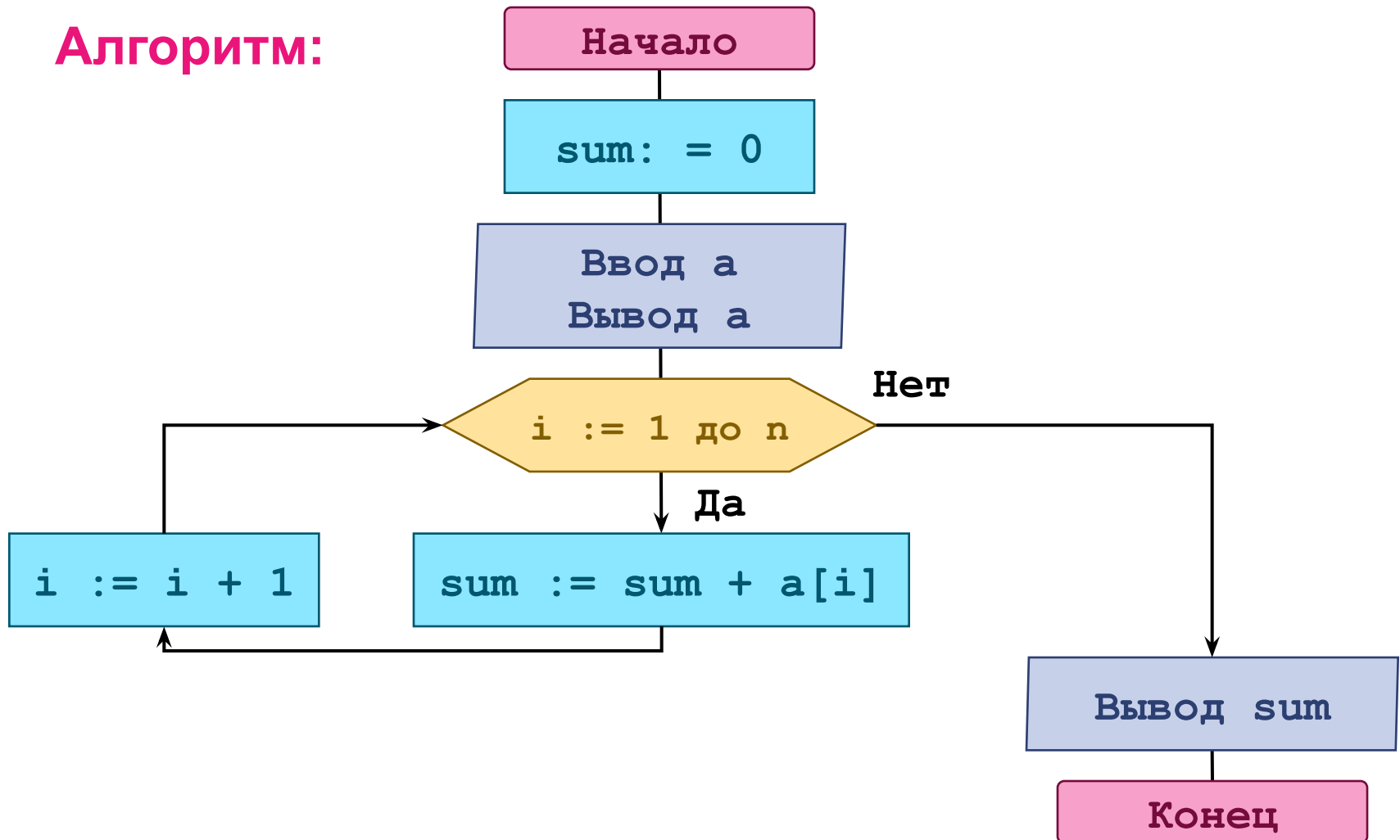
Словесный алгоритм решения задачи:

1. Задать начальное значение переменной `sum`.
2. Сформировать и вывести массив на экран.
3. Пройти весь массив, записав в `sum` сумму всех элементов.
4. Вывести результат на экран.

Обработка элементов массива

Вычисление суммы элементов массива

Алгоритм:



Обработка элементов массива

Вычисление суммы элементов массива

Трассировка:

```
sum := 0;  
For i := 1 to n do sum := sum + a[i];  
WriteLn ('Сумма элементов массива = ', sum);
```

n	i	a[i]	sum
5	—	—	0
—	1	15	15
—	2	5	20
—	3	0	20
—	4	-25	-5
—	5	10	5

Обработка элементов массива

Вычисление произведения элементов массива

Задача № 3

Дан массив 10 целых случайных чисел в интервале [0..20]. Найти произведение элементов массива с i -го по j -й (i и j вводятся с клавиатуры).

Входные данные:

			i		j		n	
z	1	2	3	4	5	6	..	10
a	0	1	15	10	8	6	..	16

Выходные данные: $pr = a[4] * a[5] * a[6]$

Словесный алгоритм решения задачи:

Какие ограничения
необходимы для i и j ?

1. Сформировать и вывести массив на экран.
2. Задать начальное значение переменных i , j и pr .
3. Пройти массив от i до j , записав в pr произведение элементов.
4. Вывести результат на экран.

Обработка элементов массива

Вычисление среднего арифметического элементов массива

Задача № 4

Сформировать массив из 10 целых случайных чисел в интервале $[-15..0]$, найти среднее арифметическое всех элементов массива.

Входные данные:

							n	
i	1	2	3	4	5	6	..	10
a	0	-1	-15	-10	-8	-6	..	-2

Выходные данные: $mid = (a[1] + a[2] + \dots + a[n]) / n$

Словесный алгоритм решения задачи:

1. Сформировать и вывести массив на экран.
2. Задать начальное значение переменной **mid**.
3. Пройти весь массив, вычислить сумму элементов массива.
4. Вычислить **mid**.
5. Вывести результат на экран.

Обработка элементов массива

Подсчет количества элементов массива

Задача № 5

Заполнить массив случайными числами в интервале $[-1..1]$ и подсчитать количество нулевых элементов.

Входные данные:

	n				
i	1	2	3	4	5
a	0	1	-1	-1	0

Выходные данные: переменная-счетчик **k**.

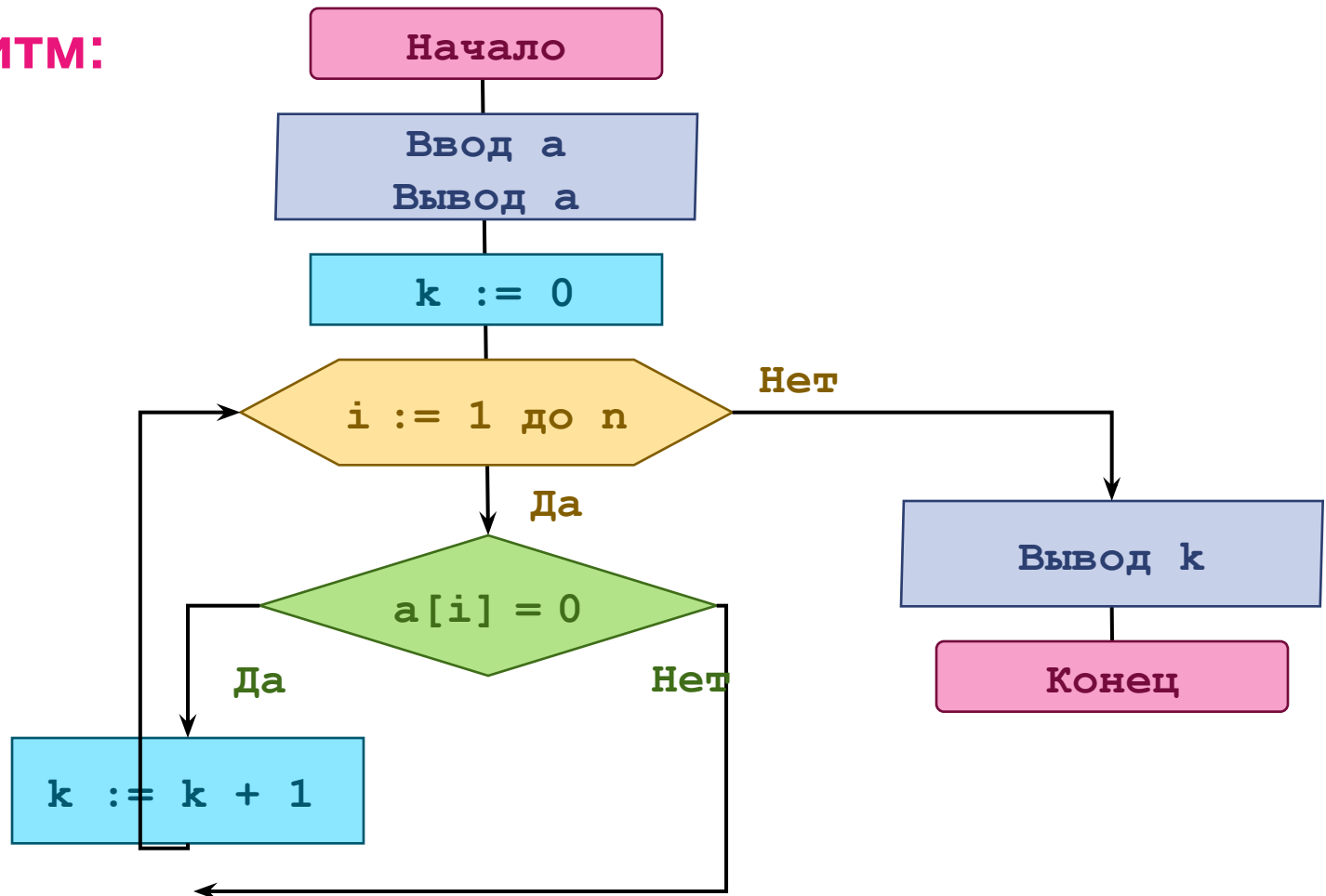
Словесный алгоритм решения задачи:

1. Сформировать и вывести массив на экран.
2. Задать начальное значение переменной-счетчика **k**.
3. Пройти весь массив. Если текущий элемент равен 0, то увеличить счетчик на 1 и перейти к следующему элементу.
4. Вывести результат на экран.

Обработка элементов массива

Подсчет количества элементов

Алгоритм:



Обработка элементов массива

Подсчет количества элементов

Трассировка:

```
k := 0;  
For i := 1 to n do  
  If a[i] = 0 then inc(k);  
WriteLn ('Количество нулевых элементов = ', k);
```

n	i	a[i]	k
5	—	—	0
—	1	0	1
—	2	1	1
—	3	-1	1
—	4	-1	1
—	5	0	2

Обработка элементов массива

Поиск максимального элемента массива

Задача № 6

Заполнить массив из 10 элементов случайными числами в интервале $[-10..10]$ и найти в нем первый (последний) максимальный элемент и его номер.

Входные данные:

									n
i	1								
a	1								
max	1								
im	1								

Выходные данные: `max = 10, im = 8.`

Обработка элементов массива

Поиск максимального элемента массива

Словесный алгоритм решения задачи:

1. Сформировать и вывести массив на экран.
2. Задать начальное значение переменных `max` и `im`.
3. Пройти массив. Если текущий элемент больше максимального, то изменить значения максимального элемента и его индекса.
4. Вывести результат на экран.

Запись на алгоритмическом языке:

начало

ввод `a`, вывод `a`

`max := a[1]`, `im := 1`

нц для `i` от 2 до `n`

если `a[i] > max`, тогда `max := a[i]`, `im := i`

кц

вывод `max`, `im`

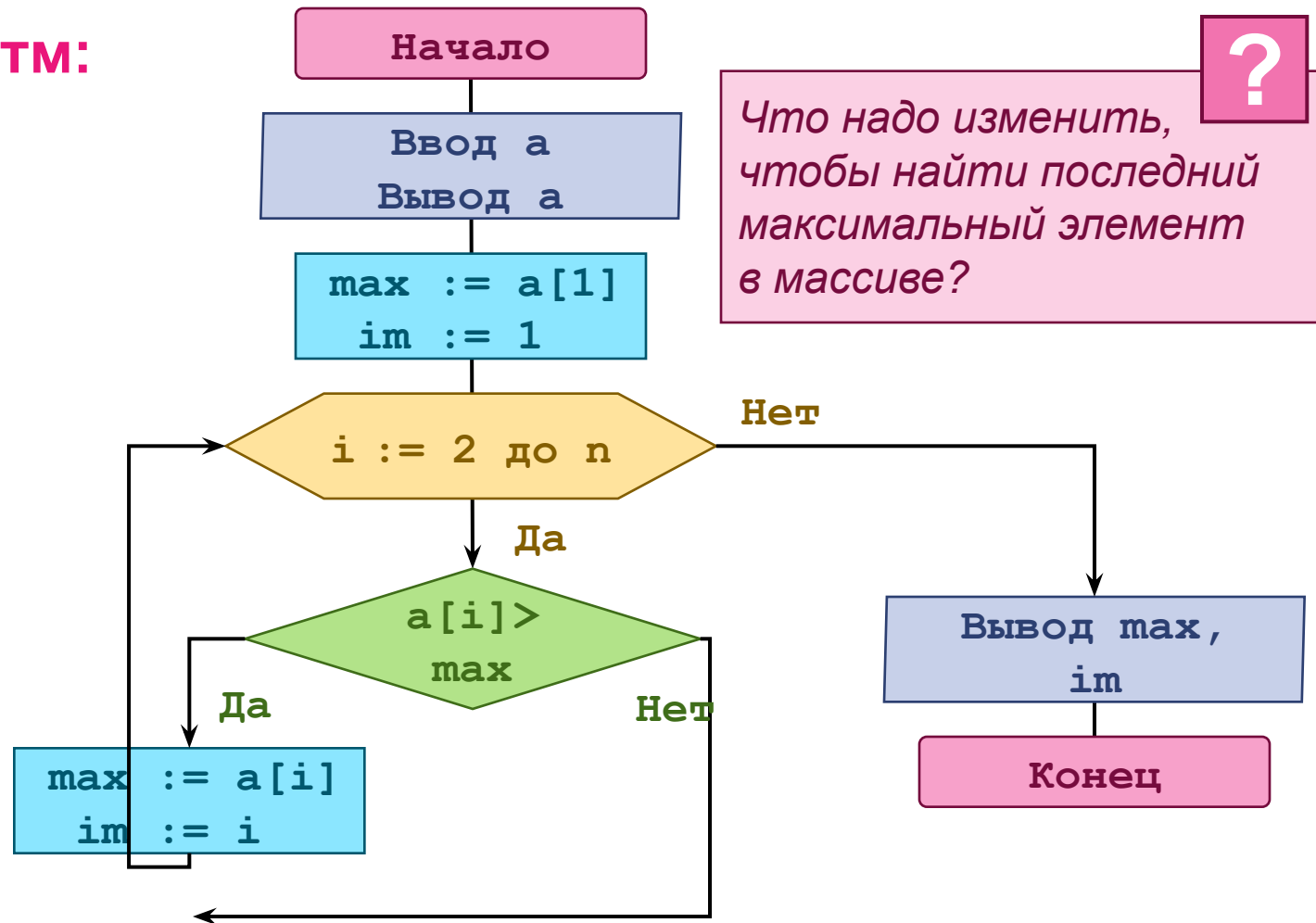
конец



Обработка элементов массива

Поиск максимального элемента

Алгоритм:



Обработка элементов массива

Поиск максимального элемента массива

Трассировка:

```
max := a[1]; im := 1;  
For i := 2 to n do  
  If a[i] > max  
  then begin  
    max := a[i];  
    im := i;  
  end;
```

n	i	a[i]	max	im
10	—	—	1	1
—	2	0	1	1
—	3	5	5	3
—	4	8	8	4
—	5	-9	8	4
—	6	9	9	6
—	7	-9	9	6
—	8	10	10	8
—	9	2	10	8
—	10	10	10	8

?

Как найти минимальный элемент в массиве?

Обработка элементов массива

Задачи для самостоятельного выполнения

Задача № 7

Заполнить массив случайными числами в интервале $[-10..10]$ и подсчитать сумму положительных элементов.

Задача № 8

Заполнить массив из 10 элементов случайными числами в интервале $[0..100]$ и подсчитать отдельно среднее значение всех элементов, которые < 50 , и среднее значение всех элементов, которые ≥ 50 .

Задача № 9

Заполнить массив случайными числами в интервале $[20..100]$ и подсчитать отдельно число четных и нечетных элементов.

Обработка элементов массива

Задачи для самостоятельного выполнения

Задача № 10

Заполнить массив случайными числами в интервале [1000..2000] и подсчитать число элементов, у которых вторая с конца цифра — четная.

Задача № 11

Заполнить массив из 15 элементов случайными числами в интервале [-20..20] и найти в нем минимальный положительный элемент и его номер.

Задача № 12

Заполнить массив из 10 элементов случайными числами в интервале [0..5] и вывести номера всех элементов, равных X.

Обработка элементов массива

Задачи для самостоятельного выполнения

Задача № 13

Даны два массива одинаковой размерности. Получить третий массив такой же размерности, каждый элемент которого равен сумме соответствующих элементов данных массивов.

Задача № 14

Заполнить массив из 10 элементов случайными числами в интервале $[0..5]$ и определить, есть ли в нем одинаковые соседние элементы.

							n	
i	1	2	3	4	5	6	..	10
a	0	1	1	2	3	3	..	5

Обязательно ли
проходить весь массив?



Обработка элементов массива

Реверс массива

Задача № 15

Переставить элементы массива в обратном порядке.

Входные данные:

										n
i	1	2	3	4	5	6	7	8	9	10
a	1	3	4	5	2	0	1	2	3	5

Выходные данные:

										n
i	1	2	3	4	5	6	7	8	9	10
a	5	3	2	1	0	2	5	4	3	1

Меняем местами элементы:

a[1] и a[10]

a[2] и a[9]

a[3] и a[8]

a[4] и a[7]

Сумма индексов всегда равна 11, или $n+1$.

Поэтому меняем $a[i]$ и $a[n+1-i]$

!

?

Обязательно ли
проходить весь массив?

Обработка элементов массива

Реверс массива

Словесный алгоритм решения задачи:

1. Сформировать и вывести массив на экран.
2. Пройти половину массива и поменять местами элементы $a[i]$ и $a[n+1-i]$.
3. Вывести результат на экран.

Запись на алгоритмическом языке:

```
начало
  ввод а, вывод а
  нц для i от 1 до n div 2
    k := a[i]
    a[i] := a[n+1-i]
    a[n+1-i] := k
  кц
  вывод а
конец
```



Зачем нужна
переменная k?



Обработка элементов массива

Поиск максимального элемента массива

Трассировка:

```
k := 0;  
For i := 1 to n div 2 do  
  begin  
    k := a[i];  
    a[i] := a[n+1-i];  
    a[n+1-i] := k ;  
  end;
```

										n
i	1	2	3	4	5	6	7	8	9	10
	1	3	4	5	2	0	1	2	3	5
1	5	3	4	5	2	0	1	2	3	1
2	5	3	4	5	2	0	1	2	3	1
3	5	3	2	5	2	0	1	4	3	1
4	5	3	2	1	2	0	5	4	3	1
5	5	3	2	1	0	2	5	4	3	1

Обработка элементов массива

Удаление элемента массива

Задача № 16

Удалить k -й элемент массива.

Входные данные:

						k					n
i	1	2	3	4	5	6	7	8	9	10	
a	1	3	4	5	2	0	1	2	3	5	

Выходные данные:

										n
i	1	2	3	4	5	6	7	8	9	10
a	5	3	2	1	0	5	4	3	1	

Словесный алгоритм решения задачи:

1. Сформировать и вывести массив на экран. Задать значение k .
2. Пройти массив, начиная с k -го элемента и присвоить каждому текущему элементу значение следующего элемента.
3. Вывести массив размером $n-1$. Последний элемент сделать 0.

Какие ограничения для k необходимо задать?



Обработка элементов массива

Вставка элемента массива

Задача № 17

Вставить число A после k -го элемента массива..

Входные данные:

$A = 13$

						k					n	$n+1$
i	1	2	3	4	5	6	7	8	9	10	11	
a	1	3	4	5	2	0	1	2	3	5		

Выходные данные:

						k					n	$n+1$
i	1	2	3	4	5	6	7	8	9	10	11	
a	5	3	2	1	0	5	13	4	3	1	5	

Обработка элементов массива

Дополнительные задачи

Задача № 18*

Переставьте в массиве первый положительный и последний отрицательный элементы.

Задача № 19*

Вставьте число k перед всеми отрицательными элементами.

Задача № 20*

Удалите из массива все элементы, содержащие в своей записи цифру 5.