

**ПОСТРОЕНИЕ БЛОК-  
СХЕМ К ЗАДАЧАМ  
ЛИНЕЙНОЙ,  
РАЗВЕТВЛЯЮЩЕЙ И  
ЦИКЛИЧЕСКОЙ  
СТРУКТУР**

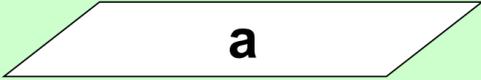
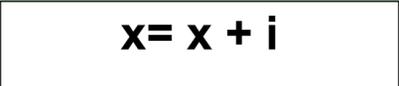
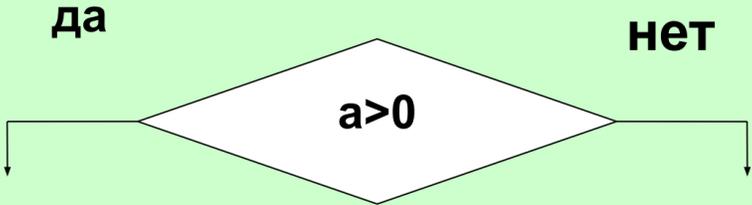
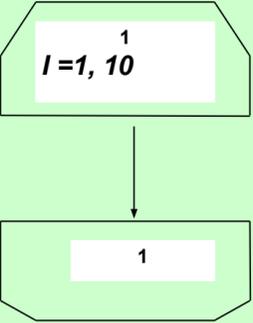
# Понятие алгоритма

**Алгоритм** - это последовательность команд, которую должен выполнить **исполнитель** для решения поставленной задачи. **Исполнителем** может быть как **человек** или какое-либо живое существо, так и **техническое устройство**. Любой исполнитель имеет систему команд - СКИ (**С**истема **К**оманд **И**сполнителя), то есть набор команд, которые он может исполнить.

Компьютер *является универсальным исполнителем*, так как его система команд (СКИ) меняется в зависимости от загруженного программного обеспечения. Алгоритм, записанный на языке программирования, называется **компьютерной программой**.

# Свойства алгоритма:

- 1) **Дискретность.** Алгоритм состоит из отдельных *команд*, причём все они входят в состав КИ.
- 2) **Конечность.** Алгоритм не может содержать бесконечное количество команд.
- 3) **Результативность.** Алгоритм должен приводить к решению поставленной задачи.

Вид блока	Назначение блока
	<p>Блок начала алгоритма</p>
	<p>Блок завершения алгоритма</p>
	<p>Блок ввода данных с клавиатуры. Внутри блока указано, в качестве примера, имя вводимой переменной.</p>
	<p>Блок вычислений. Внутри блока указано, в качестве примера, имя вычисляемой переменной.</p>
	<p>Блок условия (ветвления). Внутри блока указано, в качестве примера, проверяемое условие.</p>
	<p>Блок цикла. Внутри блока указывается количество повторений тела цикла с помощью <b>счётчика циклов</b>. В качестве примера счётчика циклов использована переменная <math>i</math>, которая изменяется от 1 до 10 с шагом 1 (по умолчанию). Таким образом, для данного примера тело цикла повторится 10 раз</p>

# Типы алгоритмов

- 1. Линейный.** Все команды алгоритма выполняются последовательно одна за другой. Пример блок-схемы линейного алгоритма.
- 2. Разветвлённый.** Содержит блок условия (ветвления) и имеет две или более ветвей. В зависимости от истинности условия выполняется одна из ветвей. Примеры блок-схемы разветвлённого алгоритма (2 примера).
- 3. Циклический.** Содержит многократно повторяющийся фрагмент - тело цикла и обеспечивает необходимое число повторений этого фрагмента. Количество повторений тела цикла не должно быть бесконечным. Примеры блок-схемы циклического алгоритма (3 примера).

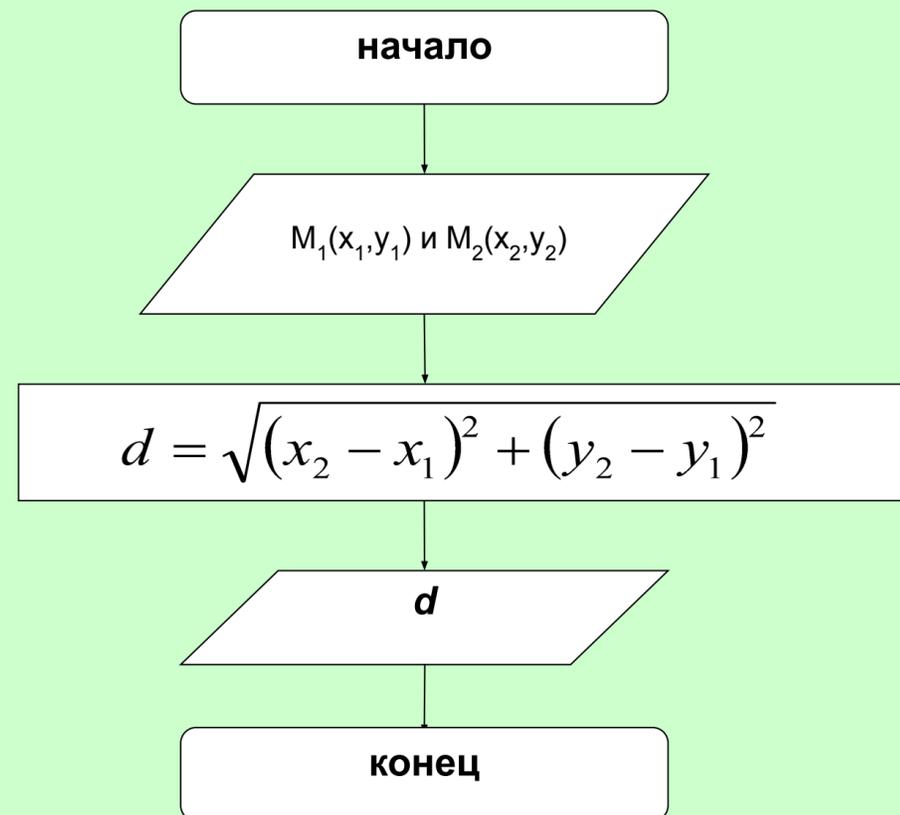
# Линейный алгоритм

**Пример 1.** Определить расстояние на плоскости между двумя точками с заданными координатами

**Этапы решения задачи:**

1. Начало;
2. Вводим координаты точек  $M_1(x_1, y_1)$  и  $M_2(x_2, y_2)$
3. Производим вычисления по формуле  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
4. Вывод результата;
5. Конец.

Составим схему алгоритма



Вернуться к [Типам алгоритмов](#)

# Разветвленный алгоритм

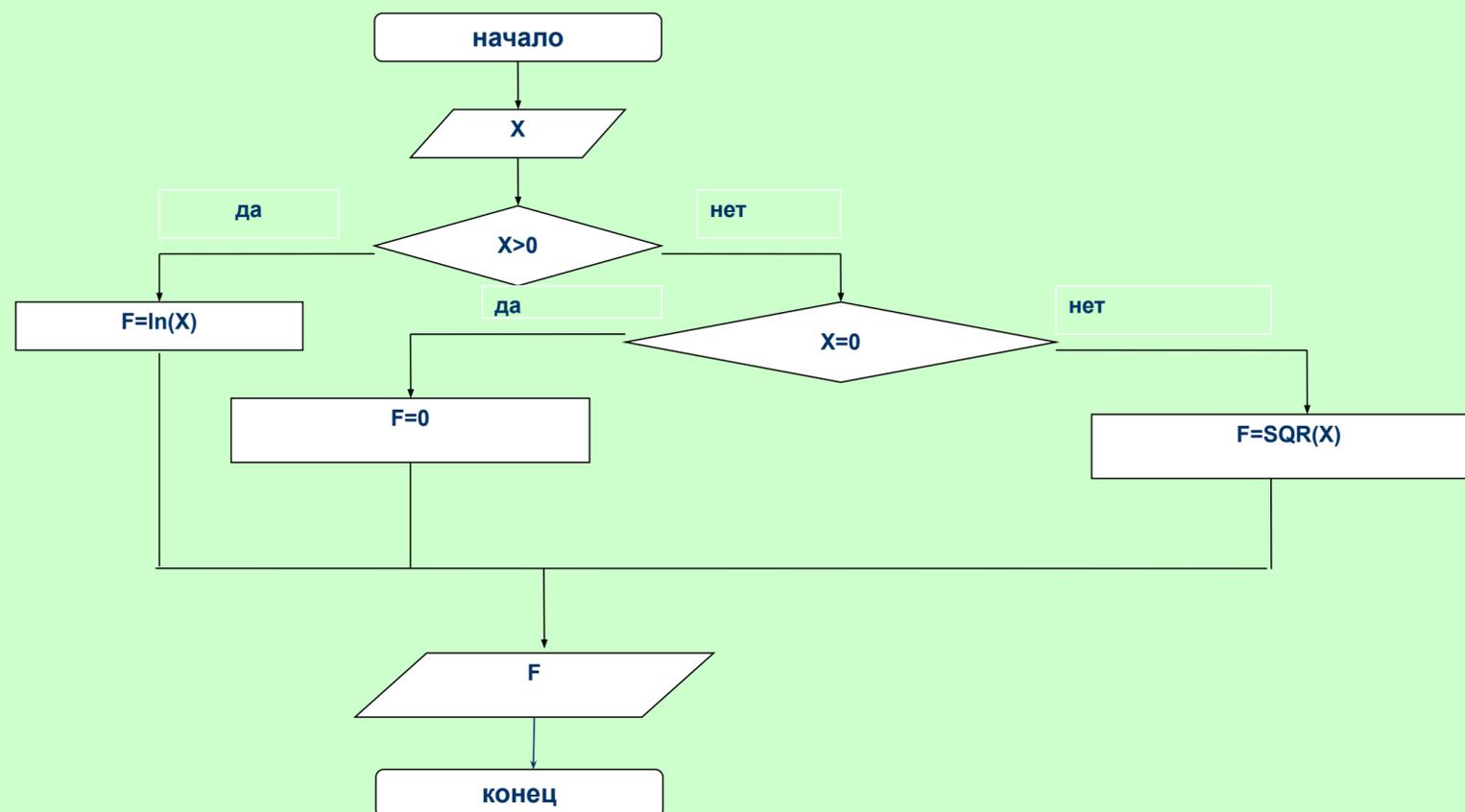
Пример 2. Дано действительное  $x$ . Вычислить  $f(x)$  для функции:

$$f(x) = \begin{cases} x^2, & \text{если } x < 0, \\ 0, & \text{если } x = 0, \\ \ln(x), & \text{если } x > 0. \end{cases}$$

**Этапы решения задачи:**

1. Начало;
2. Ввод  $X$ ;
3. Если  $X > 0$ , то  $F = \ln(X)$ ;
4. Если условие не выполняется, проверка следующего условия:
5. Если  $X = 0$ , то  $F = 0$ , иначе  $F = X^2$ ;
6. Вывод результата;
7. Конец.

**Составим схему алгоритма:**



Вернуться к [Типам алгоритма](#)

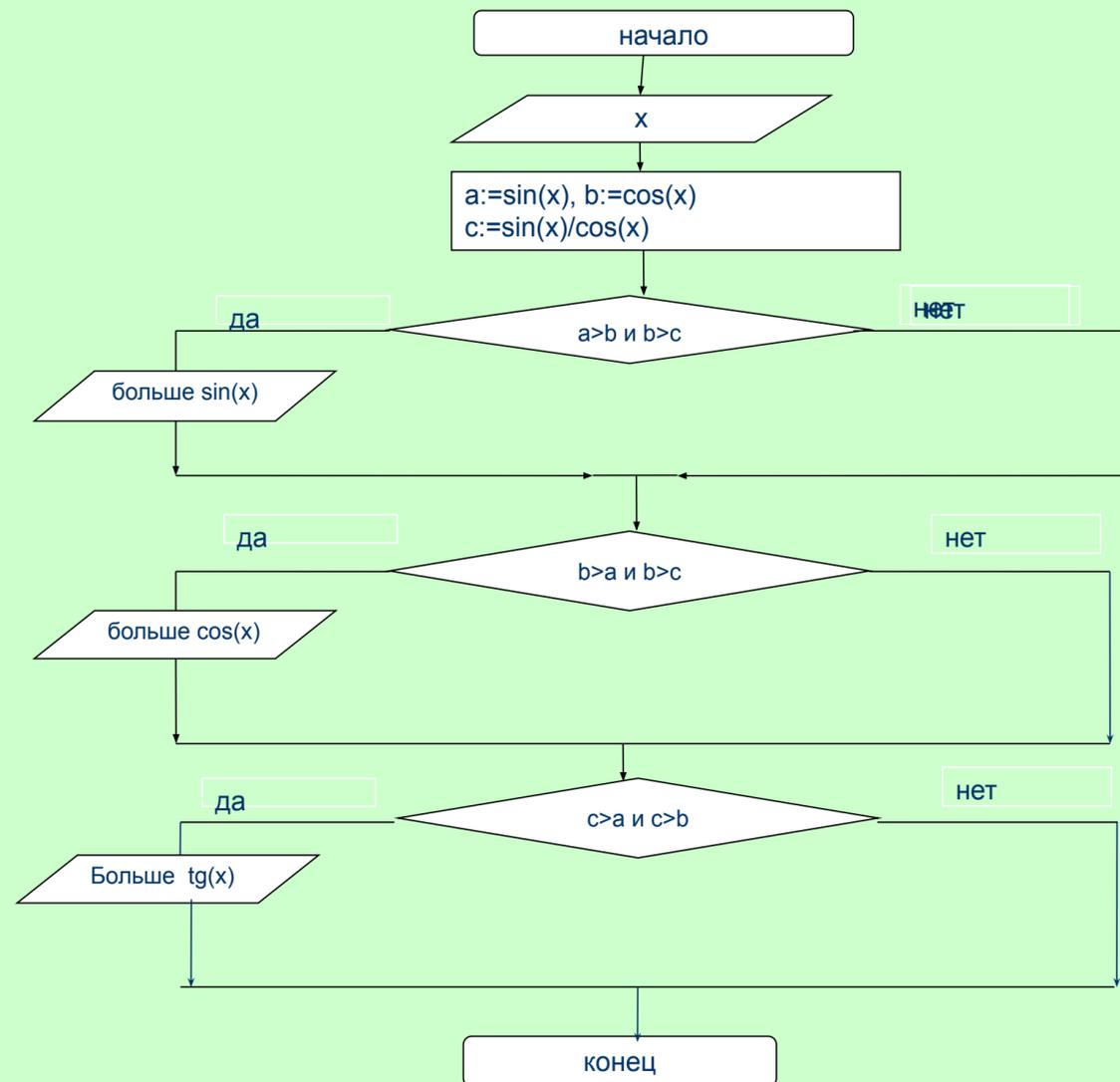
**Пример 3:** Ввести число X. Определить, что больше: SIN(X), COS(X) или TG(X).

**Этапы решения задачи:**

1. Начало;
2. Ввод x;
3. Присвоения  $a=\sin(x), b=\cos(x), c=\sin(x)/\cos(x)$ ;
4. Если  $a>b$  и  $a>c$ , то вывести сообщение «больше sin(x)»;
5. Если  $b>a$  и  $b>c$ , то вывести сообщение «больше cos(x)»;
6. Если  $c>a$  и  $c>b$ , то вывести сообщение «больше tg(x)»;
7. Конец.

Вернуться к [Типам алгоритмов](#)

Составим блок-схему алгоритма



# Циклический алгоритм

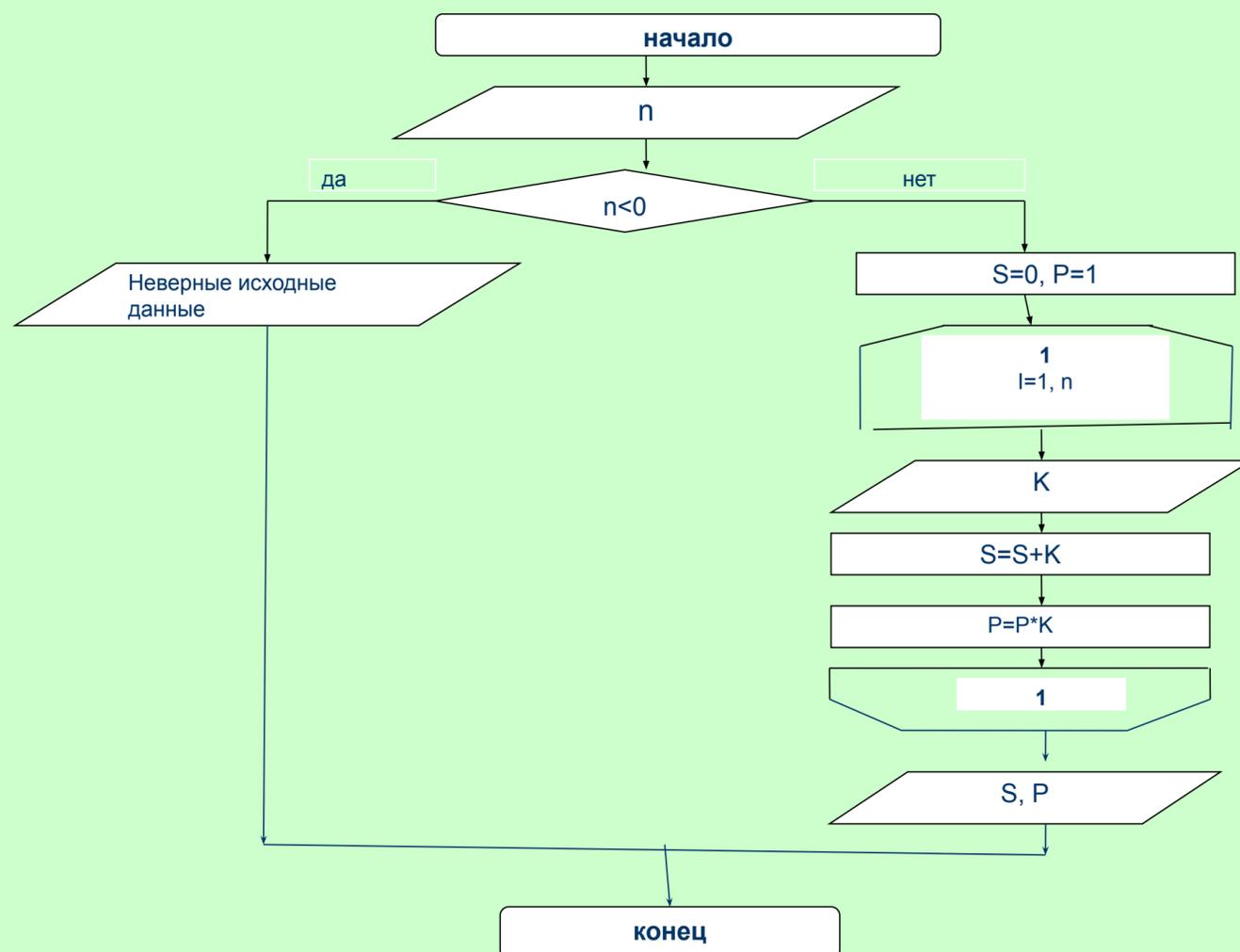
**Пример 4:** Ввести  $n$  целых чисел ( $n > 0$ ), найти их сумму, произведение.

Вернуться к [Типам алгоритмов](#)

**Этапы решения задачи:**

1. Начало;
2. Ввод  $n$ ;
3. Если  $n < 0$ , то вывод «Неверные исходные данные»;
4. Если условие не выполняется, то присвоить начальные значения  $S=0$  (первое число, к которому прибавляется следующее - это 0),  $P=1$  (первый множитель - это 1);
5. Открытие цикла  $i=$  от 1 до  $n$ ;
6. Ввод изменяемого параметра  $K$  ( $K$  - это вводимые числа);
7. Вычисление  $S, P$ ;
8. Закрытие цикла;
9. Вывод результата;
10. Конец.

Составим схему алгоритма



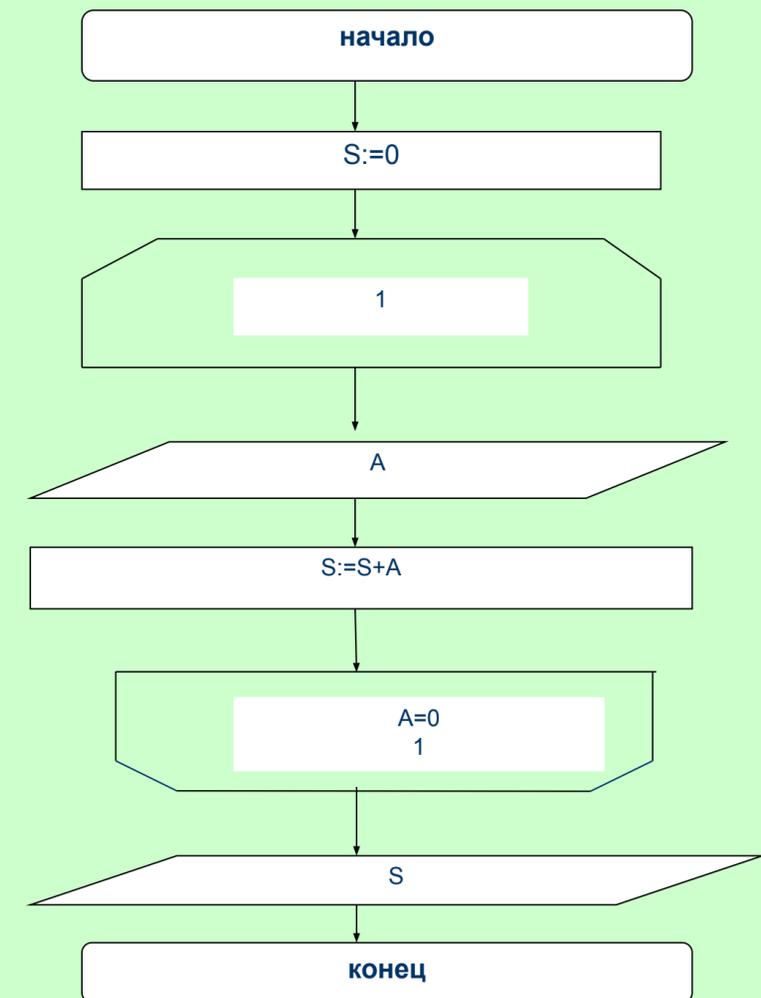
## Циклический алгоритм (с постусловием)

**Пример 5.** Найти сумму последовательности вводимых чисел. Признаком конца последовательности является ввод нуля. Вводимые слагаемые расположить в столбик, а сумму вывести. Эту задачу можно решить как с предусловием (число  $\neq 0$ ), так и с постусловием (число = 0).

**Этапы решения задачи:**

а) решение с постусловием (число = 0)

1. Начало;
2. Присваивание начального значения  $S=0$  (первое число, к которому прибавляется следующее - это 0);
3. Открытие цикла 1;
4. Ввод изменяемого параметра  $A$  ( $A$  - это вводимые числа);
5. Вычисление  $S$  (суммы чисел);
6. Закрытие цикла (условие выхода из цикла  $A=0$ );
7. Вывод результата;
8. Конец.

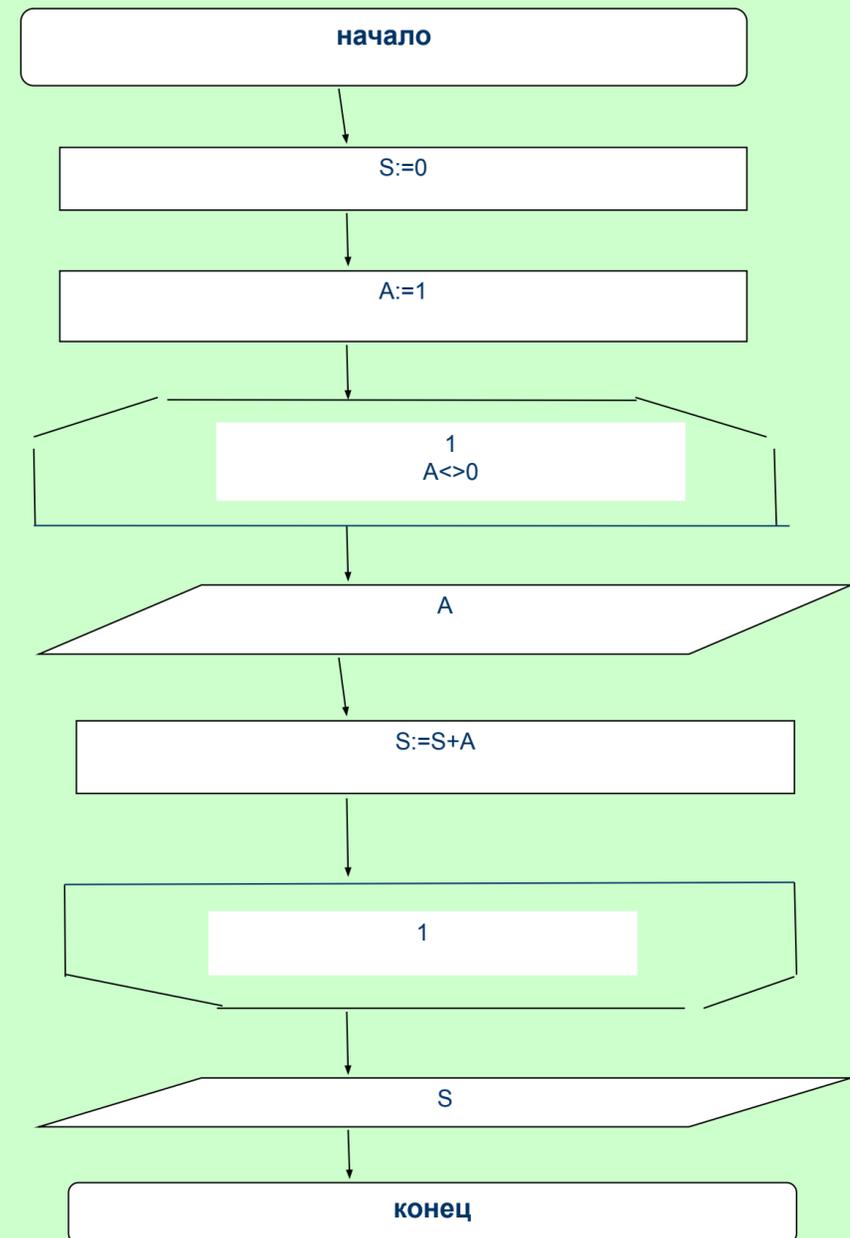


Вернуться к [Типам алгоритмов](#)

# Циклический алгоритм (с предусловием)

в) решение с предусловием (число  $\neq 0$ ):

1. Начало;
2. Присваивание начального значения  $S=0$  (первое число, к которому прибавляется следующее - это 0);
3. Ввод значения параметра  $A=1$  (начальное значение числа);
4. Начало цикла 1: условие  $A \neq 0$  ( $A$  не равно 0);
5. Вычисление  $S$  (суммы чисел) ;
6. Закрытие цикла;
7. Вывод результата;
8. Конец.



Вернуться к [Типам алгоритмов](#)