

# АЛГОРИТМ И ЕГО СВОЙСТВА

**Алгоритм** – это набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий.

Такое описание способа решения задачи, должно обладать следующими свойствами:

- Дискретность — алгоритм должен представлять процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени, то есть преобразование исходных данных в результат осуществляется во времени дискретно.
- Детерминированность (определённость). В каждый момент времени следующий шаг работы однозначно определяется состоянием системы. Таким образом, алгоритм выдаёт один и тот же результат (ответ) для одних и тех же исходных данных. С другой стороны, существуют вероятностные алгоритмы, в которых следующий шаг работы зависит от текущего состояния системы и генерируемого случайного числа. Однако при включении метода генерации случайных чисел в список «исходных данных», вероятностный алгоритм становится подвидом обычного.
- Понятность — алгоритм должен включать только те команды, которые доступны исполнителю и входят в его систему команд.
- Завершаемость (конечность) — при корректно заданных исходных данных алгоритм должен завершать работу и выдавать результат за конечное число шагов. С другой стороны, вероят-

## Алгоритм и его свойства. Способы записи алгоритмов

---

ностный алгоритм может и никогда не выдать результат, но вероятность этого равна 0.

- Массовость (универсальность). Алгоритм должен быть применим к разным наборам исходных данных.
- Результативность — завершение алгоритма определёнными результатами.

В качестве исполнителя алгоритма может выступать человек (см. рис. 1) или некоторый механизм (компьютер, токарный станок, швейная машина).

Процесс разработки алгоритма для решения задачи называется **алгоритмизацией**.



Рис. 1. Рецепт приготовления шоколадной помадки<sup>1</sup>

<sup>1</sup> Пекелис В. Д. Маленькая энциклопедия о большой кибернетике. – М.: Дет. лит., 1973, С. 396.

### История термина<sup>2</sup>

---

Слово «алгоритм» происходит от имени хорезмского учёного Абу Абдуллах Мухаммеда ибн Муса аль-Хорезми. Около 825 года он написал сочинение, в котором впервые дал описание придуманной в Индии позиционной десятичной системы счисления. Аль-Хорезми сформулировал правила вычислений в новой системе и, вероятно, впервые использовал 0 для обозначения пропущенной позиции в записи числа (её индийское название арабы перевели как *as-sifr* или просто *sifr*, отсюда такие слова, как «цифра» и «шифр»). Приблизительно в это же время индийские цифры начали применять и другие арабские учёные. В первой половине XII века книга аль-Хорезми в латинском переводе проникла в Европу. Переводчик дал ей название *Algoritmi de numero Indorum* («Алгоритмы о счёте индийском»). Латинизированное имя среднеазиатского учёного было вынесено в заглавие книги, и сегодня считается, что слово «алгоритм» попало в европейские языки именно благодаря этому сочинению. Однако вопрос о его смысле длительное время вызывал ожесточённые споры. На протяжении многих веков происхождению слова давались самые разные объяснения.



Одни выводили *algorism* из греческих *algiros* (больной) и *arithmos* (число). Из такого объяснения не очень ясно, почему числа именно «больные». Или же лингвистам больными казались люди, имеющие несчастье заниматься вычислениями? Своё объяснение предлагал и энциклопедический словарь Брокгауза и Ефрона. В нём алгоритм (в то время писали «алгорифм») производится «от арабского слова Аль-Горетм, то есть корень».

Перевод сочинения аль-Хорезми стал первой ласточкой, и в течение нескольких следующих столетий появилось множество

---

<sup>2</sup> Википедия. Статья «Аль-Хорезми»

других трудов, посвящённых всё тому же вопросу — обучению искусству счёта с помощью цифр. Слово *algorism* (или *algorismus*), неизменно присутствовавшее в названиях математических сочинений, обрело значение способа выполнения арифметических действий посредством арабских цифр.

Алгоритм – это искусство счета с помощью цифр.

Именно в таком значении оно вошло во многие европейские языки.

Но постепенно значение слова расширялось. Учёные начинали применять его не только к сугубо вычислительным, но и к другим математическим процедурам. Например, около 1360 г. французский философ Николай Орем (*Nicolaus Oresme*, 1323/25-1382) написал математический трактат *Algorismus proportionum* («Вычисление пропорций»), в котором впервые использовал степени с дробными показателями и фактически вплотную подошёл к идее логарифмов.

В 1684 году Готфрид Лейбниц в сочинении *Nova Methodvs pro maximis et minimis, itemque tangentibus...* впервые использовал слово «алгоритм» (*Algorithmo*) в ещё более широком смысле: как систематический способ решения проблем дифференциального исчисления.

Однако потребовалось ещё почти два столетия, чтобы все старинные значения слова вышли из употребления. Этот процесс можно проследить на примере проникновения слова «алгоритм» в русский язык.

Историки датируют 1691 годом один из списков древнерусского учебника арифметики, известного как «Счётная мудрость». Это сочинение известно во многих вариантах (самые ранние из них почти на сто лет старше) и восходит к ещё более древним рукописям XVI в. По ним можно проследить, как знание арабских цифр и правил действий с ними постепенно распространялось на Руси. Полное название этого учебника — «Сия книга, глаголемая по еллински и по гречески арифметика, а по немецки алгоризма, а по русски цифирная счётная мудрость».

Таким образом, слово «алгоритм» понималось первыми русскими математиками так же, как и в Западной Европе. Однако его не было ни в знаменитом словаре В.И. Даля, ни спустя сто лет в «Толковом словаре русского языка» под редакцией Д. Н. Ушакова (1935 г.). Зато слово «алгорифм» можно найти и в популярном до-революционном Энциклопедическом словаре братьев Гранат, и в первом издании Большой советской энциклопедии (БСЭ), изданном в 1926 г. И там, и там оно трактуется одинаково: как правило, по которому выполняется то или иное из четырёх арифметических действий в десятичной системе счисления.

Однако к началу XX века для математиков слово «алгоритм» уже означало любой арифметический или алгебраический процесс, выполняемый по строго определённым правилам, и это объяснение также даётся в следующих изданиях БСЭ.

Алгоритмы становились предметом всё более пристального внимания учёных, и постепенно это понятие заняло одно из центральных мест в современной математике. Что же касается людей, от математики далёких, то к началу сороковых годов XX в. это слово они могли услышать разве что во время учёбы в школе, в сочетании «алгоритм Евклида». Несмотря на это, алгоритм всё ещё воспринимался как термин сугубо специальный, что подтверждается отсутствием соответствующих статей в менее объёмных изданиях. В частности, его нет даже в десяти томной Малой советской энциклопедии (1957 г.), не говоря уже об одностомных энциклопедических словарях. Но зато спустя десять лет, в третьем издании Большой советской энциклопедии (1969 г.) алгоритм уже характеризуется как одна из основных категорий математики, «не обладающих формальным определением в терминах более простых понятий, и абстрагируемых непосредственно из опыта». Как мы видим, отличие даже от трактовки первым изданием БСЭ разительное! За сорок лет алгоритм превратился в одно из ключевых понятий математики, и признанием этого стало включение слова уже не в энциклопедии, а в словари. Например, оно присутствует в

## Алгоритм и его свойства. Способы записи алгоритмов

---

академическом «Словаре русского языка» (1981 г.) именно как термин из области математики.

Одновременно с развитием понятия алгоритма постепенно происходила и его экспансия из чистой математики в другие сферы. Вообще можно сказать, что его сегодняшняя известность напрямую связана со степенью распространения компьютеров.

Например, в третьем томе «Детской энциклопедии» (1959 г.) о вычислительных машинах говорится немало, но они ещё не стали чем-то привычным и воспринимаются скорее как некий атрибут светлого, но достаточно далёкого будущего. Соответственно и алгоритмы ни разу не упоминаются на её страницах. Но уже в начале 70-х гг. XX столетия, когда компьютеры перестали быть экзотической диковинкой, слово «алгоритм» стремительно входит в обиход. Это чутко фиксируют энциклопедические издания. В «Энциклопедии кибернетики» (1974 г.) в статье «Алгоритм» он уже связывается с реализацией на вычислительных машинах, а в «Советской военной энциклопедии» (1976 г.) даже появляется отдельная статья «Алгоритм решения задачи на ЭВМ». Теперь компьютер стал неотъемлемым атрибутом нашей жизни, компьютерная лексика становится всё более привычной. Слово «алгоритм» уверенно шагнуло даже в разговорную речь, и сегодня мы нередко встречаем в газетах и слышим в выступлениях политиков выражения вроде «алгоритм поведения», «алгоритм успеха» или даже «алгоритм предательства». Академик Н. Н. Моисеев назвал свою книгу «Алгоритмы развития», а известный врач Н.М. Амосов — «Алгоритм здоровья» и «Алгоритмы разума». А это означает, что слово живёт, обогащаясь всё новыми значениями и смысловыми оттенками.

### Классификация алгоритмов

---

Алгоритмы в зависимости от порядка действий исполнителя подразделяются следующим образом:

- **Линейный алгоритм** — набор команд (указаний), выполняемых последовательно во времени друг за другом.
- **Разветвляющийся алгоритм** — алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько параллельных ветвей алгоритма.
- **Циклический алгоритм** — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Цикл программы — последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторого условия.



# СПОСОБЫ ЗАПИСИ АЛГОРИТМОВ

Существуют различные способы записи алгоритмов:

- Словесно-формульная,
- Блок-схема,
- Псевдокод (алгоритмические языки).

## Словесно-формульная запись алгоритма

---

Для исполнителя-человека алгоритм может быть представлен предложениями обычного текста с использованием в случае необходимости математической или другой символики. Такой способ записи алгоритма называют **словесно-формульным**.

Пример 1. Рецепт приготовления шоколадной помадки, рис. 1.

Пример 2. Алгоритм поиска вещественных корней квадратного уравнения:  $ax^2 + bx + c = 0$ ,  $a \neq 0$ :

1. Задать значения коэффициентов  $a$ ,  $b$  и  $c$ .
2. Вычислить дискриминант  $D = b^2 - 4ac$ .
3. Проверить условие существования вещественных корней  $D \geq 0$ . Если  $D \geq 0$  выполнено, то перейти к шагу 5. Если не выполнено (т.е.  $D < 0$ ), то перейти к шагу 4.
4. Дать ответ – «вещественных корней нет». Перейти к шагу 8.
5. Проверить условие существования двух различных корней  $D > 0$ . Если условие выполнено, то перейти к шагу 7, иначе – перейти к шагу 6.
6. Вычислить корень уравнения  $x = -b/2a$ . Дать ответ – «уравнение имеет один корень  $x$ ». Перейти к шагу 8.
7. Вычислить корни уравнения  $x_1 = \frac{-b+\sqrt{D}}{2a}$ ,  $x_2 = \frac{-b-\sqrt{D}}{2a}$ . Дать ответ – «уравнение имеет два корня  $x_1, x_2$ ».
8. Закончить вычисления.

### Блок-схема

---






Наглядным и удобным способом изображения алгоритмов для исполнителя-человека являются блок-схемы.

**Блок-схема** алгоритма – это его графическое изображение в виде схемы связанных между собой с помощью линий перехода блоков – специальных графических объектов, каждый из которых соответствует определенным шагам алгоритма.

Внутри блока дается описание соответствующих действий. Блоки, как правило, располагаются сверху вниз. Линии соединения отдельных блоков показывают направление процесса обработки в схеме. Стрелки на соединяющих линиях не ставятся при направлениях сверху вниз и слева направо; противоположные направления показывают стрелкой на линии.

При изображении блок-схем используются следующие типы блоков:

---

	Начало и конец алгоритма
	Блок ввода-вывода информации
	Блок обработки. Внутри блока записываются формулы, обозначения операций и функций
	Блок выбора. Внутри блока записываются условия выбора следующего действия алгоритма
	Блок заголовка цикла с параметром

---



Соединительный блок

---

Существуют три основные разновидности алгоритмических действий, сочетанием которых можно описать любой алгоритм:

- линейная последовательность (следование),
- выбор (разветвление),
- повторение (цикл).

**Линейная последовательность (следование)** – это набор действий, выполняемых последовательно друг за другом. На блок-схеме ему соответствует несколько расположенных друг за другом блоков обработки и/или блоков ввода-вывода (рис. 1).

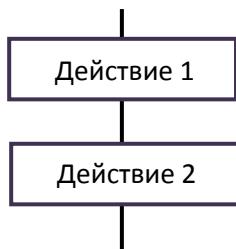


Рис. 1

**Выбор (разветвление)** – это набор действий, начинающийся с проверки некоторого условия. Результатом проверки может быть один из двух исходов: условие выполнено («ДА») или условие не выполнено («НЕТ»). Каждому из этих исходов могут соответствовать некоторые действия (рис. 2) или отсутствие каких-либо действий для одного из исходов (рис. 3). Обычно условие формулируется в виде логического выражения, вычисление которого в зависимости от результатов предшествующих действий дает значение **истина** («ДА») или **ложь** («НЕТ»).

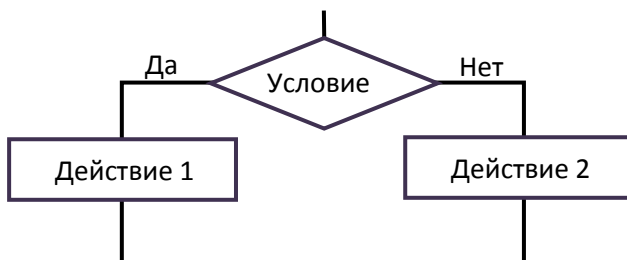


Рис. 2

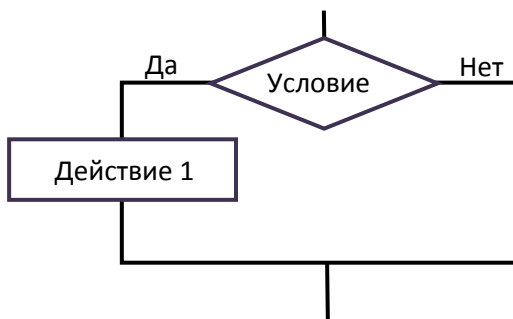


Рис. 3

**Повторение** (цикл) – это набор действий, которые могут выполняться многократно. Этот набор включает проверку условия, определяющего количество повторений цикла. Различают три варианта циклов:

- цикл с предусловием,
- цикл с постусловием,
- цикл с параметром.

## Алгоритм и его свойства. Способы записи алгоритмов

Цикл с предусловием («цикл-пока») (рис. 4) проверка условия является первым выполняемым действием; если результат проверки – «ДА», то выполняются другие действия цикла (тело цикла) и затем вновь проверка условия и т. д.; если результат проверки условия – «НЕТ», то цикл заканчивается. Действия цикла, следующие за проверкой условия, могут ни разу не выполняться, если результат первой проверки условия – «НЕТ».

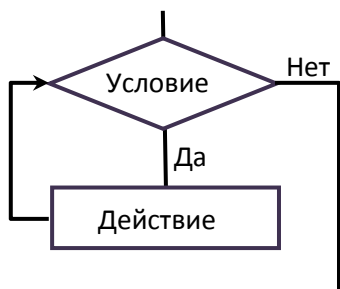


Рис. 4

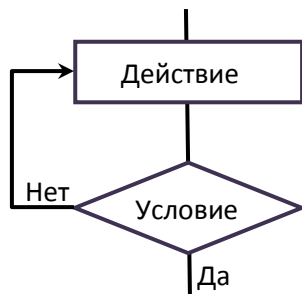


Рис. 5

В цикле с постусловием («Цикл-до») (рис. 5) проверка условия является последним выполняемым в составе цикла действием; если результат проверки – «ДА», то цикл заканчивается; если результат проверки условия – «НЕТ», то вновь выполняются все предшествующие действия цикла и т. д. В отличие от цикла с предусловием все действия в цикле с постусловием всегда, по крайней мере, один раз выполняются.

Для того чтобы циклы с пред- или постусловием заканчивались после какого-то количества повторений, действия внутри циклов должны изменять некоторые величины, влияющие на результат проверки условия. Само количество повторений таких циклов может быть заранее, до начала их выполнения, неизвестно. В тех случаях, когда число повторений известно заранее или определяется до начала цикла, может быть использован цикл с параметром. Такой цикл начинается с заголовка, содержащего параметр цикла с указанием его начального и конечного значений; за заголовком

## Алгоритм и его свойства. Способы записи алгоритмов

следуют действия, повторяемые в этом цикле. При каждом повторении цикла параметр возрастает (или убывает) на некоторую постоянную величину (шаг), изменяясь от начального до конечного значения. Набор действий, образующих цикл с параметром, показан на блок-схеме на рис. 6.

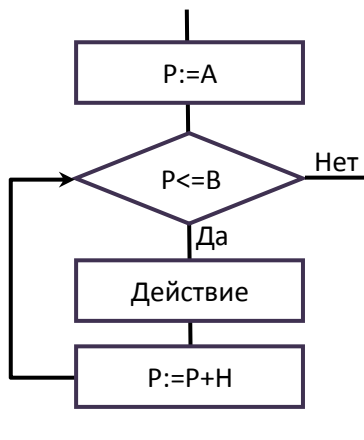


Рис. 6

Для упрощения блок-схем при изображении цикла с параметром можно использовать специальный блок-заголовок цикла с параметром. Такой вариант показан на рис. 7.

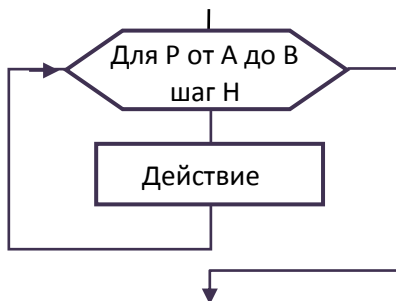


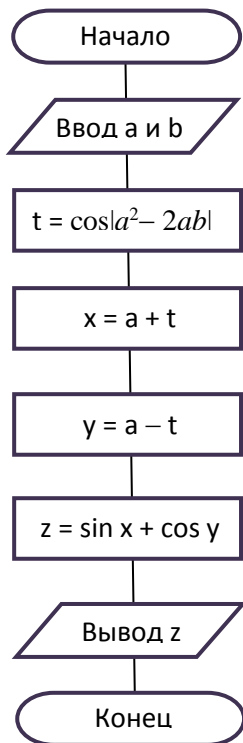
Рис. 7

### Примеры записи алгоритма в виде блок-схем

---

#### Пример 1

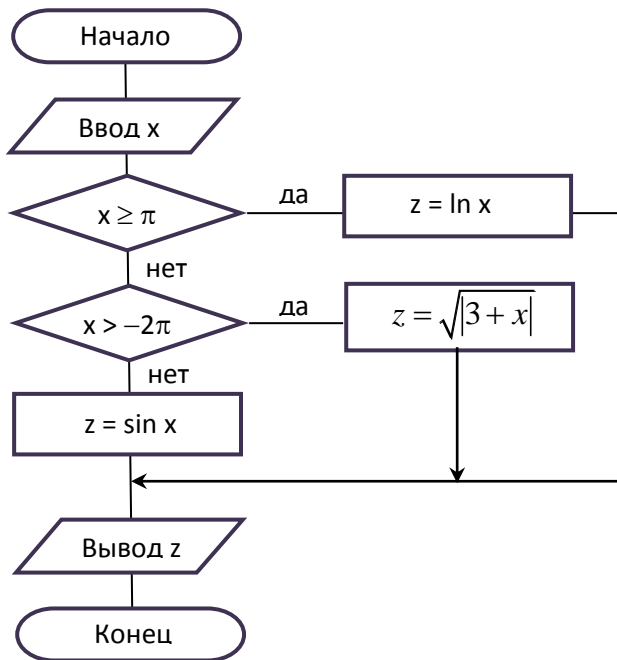
Задавая  $a$  и  $b$ , вычислить значения функции  $z = \sin x + \cos y$ , где  $x = a + \cos|a^2 - 2ab|$ ,  $y = a - \cos|a^2 - 2ab|$ .



Пример 2

Составить алгоритм вычисления функции

$$z = \begin{cases} \ln x, & \text{если } x \geq \pi, \\ \sqrt{|3+x|}, & \text{если } -2\pi < x < \pi, \\ \sin x, & \text{если } x \leq -2\pi. \end{cases}$$



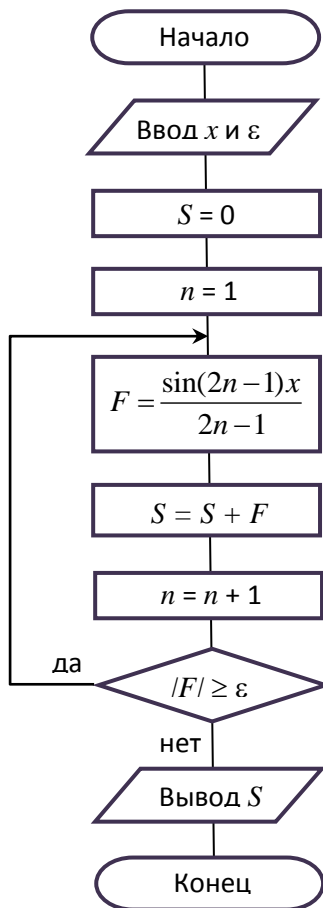


### Пример 3

Составить алгоритм вычисления суммы ряда:

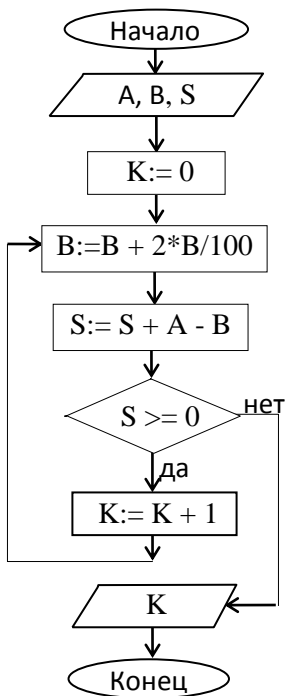
$$S = \sin x + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots = \sum_{n=1}^{\infty} \frac{\sin(2n-1)x}{2n-1}$$

с точностью до члена ряда, меньшего  $\varepsilon$ , для заданного значения  $x$ .



### Пример 4

У студента имеются накопления в сумме  $S$  рублей. Ежемесячная стипендия составляет  $A$  рублей, расходы на проживание превышают стипендию и составляют  $B$  рублей в месяц накануне начала учебы. Рост цен ежемесячно увеличивает расходы на 2% по сравнению с расходами предыдущего месяца. Определить количество месяцев, которые может прожить студент, используя только накопления и стипендию.



Ввод размера стипендии, расходов и накоплений.

Начальное значение для количества месяцев.

Расходы за месяц составят.

При этом остаток суммы на начало следующего месяца будет равен.

Хватит ли денег?

Можно прожить еще месяц.

Вывод количества месяцев, которые можно прожить только за счет накоплений и стипендии.

## Алгоритм и его свойства. Способы записи алгоритмов

---

Графическое изображение алгоритма широко используется перед программированием задачи вследствие его наглядности, так как зрительное восприятие обычно облегчает процесс написания программы, ее корректировки при возможных ошибках, осмысливание процесса обработки информации.

### Запись алгоритма с помощью алгоритмических языков

---

Алгоритмы, предназначенные для исполнения на ЭВМ, описывают специально для этого созданных формальных языках, которые называют **алгоритмическими** или **языками программирования**. Такую запись алгоритма называют **программой**.

### Примеры записи алгоритма в виде программ

---

#### Пример 1

Задавая  $a$  и  $b$ , вычислить значения функции  $z = \sin x + \cos y$ , где  $x = a + \cos|a^2 - 2ab|$ ,  $y = a - \cos|a^2 - 2ab|$ .

#### Паскаль

```
var a, b, x, y, t, z : real;
begin
  readln(a, b); {ввод a и b}
  t:= cos(abs(a*a-2*a*b));
  x:=a + t; {вычисление значения x}
  y:=a - t; {вычисление значения y}
  z:=sin(x) + cos(y); {вычисление значения z}
  write(z) {вывод результата}
end.
```

Алгоритмический язык
<pre>алг нач вещ a, b, x, y, t, z   ввод a, b   t:= cos(abs(a*a-2*a*b))   x:=a + t   y:=a - t   z:=sin(x) + cos(y)   вывод z кон</pre>
Бейсик
<pre>INPUT A, B T = COS(ABS(A*A-2*A*B)) X=A + T 'вычисление значения X Y=A - T 'вычисление значения Y Z=SIN(X) + COS(Y) 'вычисление значения z PRINT Z END</pre>
Си
<pre>#include&lt;stdio.h&gt; #include&lt;math.h&gt; void main(void) { float a, b, x, y, t, z;   scanf("%f %f",&amp;a,&amp;b); // ввод a и b   t = cos(fabs(a*a-2*a*b));   x = a + t; // вычисление значения x   y = a - t; // вычисление значения y   z = sin(x) + cos(y); // вычисление значения z   printf("%f ", z); // вывод результата }</pre>

### Пример 2

Составить алгоритм вычисления функции

$$z = \begin{cases} \ln x, & \text{если } x \geq \pi, \\ \sqrt{|3+x|}, & \text{если } -2\pi < x < \pi, \\ \sin x, & \text{если } x \leq -2\pi. \end{cases}$$

#### Паскаль

```
var x, z : real;
begin
  readln(x); {ввод x}
  if x >= pi
  then z := ln(x)
  else if x >= -2*pi
       then z := sqrt(abs(3+x))
       else z := sin(x);
  write(z) {вывод результата}
end.
```

#### Алгоритмический язык

```
алг
нач
вещ a, b, x, y, t, z
ввод x
если x >= pi то z := ln(x)
  иначе
    если x >= -2*pi то z := корень(|3 + x|)
    иначе z := sin(x)
конец если
вывод z
кон
```

### Бейсик

```
INPUT X
IF X>=PI
THEN
    Z= LN(X)
ELSE
    IF X >= -2*PI
    THEN
        Z= SQR(ABS(3+X))
    ELSE
        Z= SIN(X)
    ENDIF
ENDIF
PRINT Z
END
```

### Си

```
#include<stdio.h>
#include<math.h>
void main(void)
{
    float x, z, pi = 3.1415;
    scanf("%f",&x); // ввод x
    if (x>=pi) z = log(x);
    else if (x >= -2*pi) z= sqrt(fabs(3+x));
    else z= sin(x);
    printf("%f ", z); // вывод результата
}
```

### Пример 3

Составить алгоритм вычисления суммы ряда:

$$S = \sin x + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots = \sum_{n=1}^{\infty} \frac{\sin(2n-1)x}{2n-1}$$

с точностью до члена ряда, меньшего  $\varepsilon$ , для заданного значения  $x$ .

#### Паскаль

```
var x, e, f, s: real;
    n: integer;
begin
  readln(x, e); {ввод x}
  s:= 0; {начальное значение суммы}
  n:=1; {номер слагаемого}
  f:=sin(x); {значение первого слагаемого}
  while (abs(f)>=e) do
  begin
    s:=s+f; {добавляем слагаемое к сумме}
    n:=n+1; {номер следующего слагаемого}
    f:= sin(2*n - 1)*x/ (2*n - 1); {добавляем слагаемое к сумме}
  end;
  write(s) {вывод результата}
end.
```

#### Алгоритмический язык

```
алг
нач
вещ x, e, f, s
целое n
ввод x, e
s:= 0 {начальное значение суммы}
n:=1 {номер слагаемого}
f:=sin(x) {значение первого слагаемого}
пока abs(f)>=e
```

## Алгоритм и его свойства. Способы записи алгоритмов

```
s:=s+f {добавляем слагаемое к сумме}
n:=n+1 {номер следующего слагаемого}
f:= sin(2*n - 1)*x/ (2*n - 1) {добавляем слагаемое к сумме}
конец пока
вывод s
конец
```

### Бейсик

```
INPUT X, E
S= 0 'начальное значение суммы
N=1 'номер слагаемого
F=SIN(X) 'значение первого слагаемого
WHILE (ABS(F)>=E)
  S=S+F 'добавляем слагаемое к сумме
  N=N+1 'номер следующего слагаемого
  F= SIN(2*N - 1)*X/ (2*N - 1) 'добавляем слагаемое к сумме
WEND
PRINT "S = ", S
END
```

### Си

```
#include<stdio.h>
#include<math.h>
void main(void)
{ float x, e, s, f; int n;
  scanf("%f %f",&x,&e); // ввод
  s= 0; //начальное значение суммы
  n=1; //номер слагаемого
  f=sin(x); //значение первого слагаемого
  while (abs(f)>=e)
  { s=s+f; //добавляем слагаемое к сумме
    n=n+1; //номер следующего слагаемого
    f= sin(2*n - 1)*x/ (2*n - 1); //добавляем слагаемое к сумме
  }
  printf(" S = %f ", s); // вывод результата
}
```



### Пример 4

У студента имеются накопления в сумме  $S$  рублей. Ежемесячная стипендия составляет  $A$  рублей, расходы на проживание превышают стипендию и составляют  $B$  рублей в месяц накануне начала учебы. Рост цен ежемесячно увеличивает расходы на 2% по сравнению с расходами предыдущего месяца. Определить количество месяцев, которые может прожить студент, используя только накопления и стипендию.

#### Паскаль

```
var a, b, s: real;
    k: integer;
begin
  readln(a, b, s); {ввод исходных данных}
  k:= 0; {начальное значение количества месяцев}
  b:=b+b*0.02;{расходы первого месяца }
  while (s+a>=b) do {накоплений и стипендии хватает еще на один
                    месяц}
  begin
    k:=k+1; {еще один месяц }
    s:=s+a-b; {остаток суммы на конец месяца}
    b:=b+b*0.02; {расходы для следующего месяца }
  end;
  write(k) {вывод результата}
end.
```

#### Алгоритмический язык

```
алг
нач
вещ a, b, s
целое k
ввод a, b, s
k:= 0 {начальное значение количества месяцев}
b:=b+b*0.02 {расходы первого месяца }
пока (s+a>=b) {накоплений и стипендии хватает еще на один
```

## Алгоритм и его свойства. Способы записи алгоритмов

```
        месяц}
    k:=k+1 {еще один месяц }
    s:=s+a-b {остаток суммы на конец месяца}
    b:=b+b*0.02 {расходы для следующего месяца }
конец пока
вывод k
кОН
```

### Бейсик

```
INPUT X, E
K= 0 'начальное значение количества месяцев
B=B+B*0.02 'расходы первого месяца
WHILE (S+A>=B)
    K=K+1 'еще один месяц
    S=S+A-B 'остаток суммы на конец месяца
    B=B+B*0.02 'расходы для следующего месяца
WEND
PRINT K
END
```

### Си

```
#include<stdio.h>
#include<math.h>
void main(void)
{ float x, e, s, f; int n;
  scanf("%f %f",&x,&e); // ввод
  k= 0; //начальное значение количества месяцев
  b=b+b*0.02; //расходы первого месяца
  while (s+a>=b)
  { k=k+1; //еще один месяц
    s=s+a-b; //остаток суммы на конец месяца
    b=b+b*0.02; //расходы для следующего месяца
  }
  printf("%d ", k); // вывод результата
}
```

### ЛИТЕРАТУРА

1. Костюк Ю.Л., Фукс И.Л. Основы разработки алгоритмов. – М.: Бином. Лаборатория знаний, 2010.
2. Криницкий А.Н. Алгоритмы вокруг нас. – М.: Наука, 1984.
3. Столяр С.Е., Владыкин А.А. Информатика. Представление данных и алгоритмы. – М.: Бином. Лаборатория знаний, 2007.
4. Паронджанов В. Учись писать, читать и понимать алгоритмы. – М.: Изд-во ДМК Пресс, 2012.
5. Князева М.Д. Алгоритмика. От алгоритма к программе. – Изд-во КУДИЦ-Образ, 2006.

### Интернет-ресурсы

1. Алгоритмизация  
<http://www.youtube.com/watch?v=uw729TH7amg>
2. Алгоритмы в программировании. Свойства алгоритмов:  
[https://www.youtube.com/watch?v=PFKf\\_ydгEPY](https://www.youtube.com/watch?v=PFKf_ydгEPY)
3. Блок-схемы в программировании:  
<https://www.youtube.com/watch?v=1-KyGz8hk4I>
4. Алгоритмы и алгоритмические языки:  
<https://www.youtube.com/watch?v=cX247ZSE-Ws>
5. Статья в ВИКИВЕРСИТЕТ. Теория алгоритмов.  
[http://ru.wikiversity.org/wiki/Теория\\_алгоритмов](http://ru.wikiversity.org/wiki/Теория_алгоритмов)
6. Статья в энциклопедии информационной безопасности. Алгоритм. <http://www.wikisec.ru/index.php?title=Алгоритм>

## ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Изобразите в виде блок-схемы следующий алгоритм:

Шаг 1. Ввести значения  $t, z$ .

Шаг 2. Проверить условие  $t \geq z$ . Если условие выполняется, то перейти к шагу 4. Если условие не выполняется, то перейти к шагу 3.

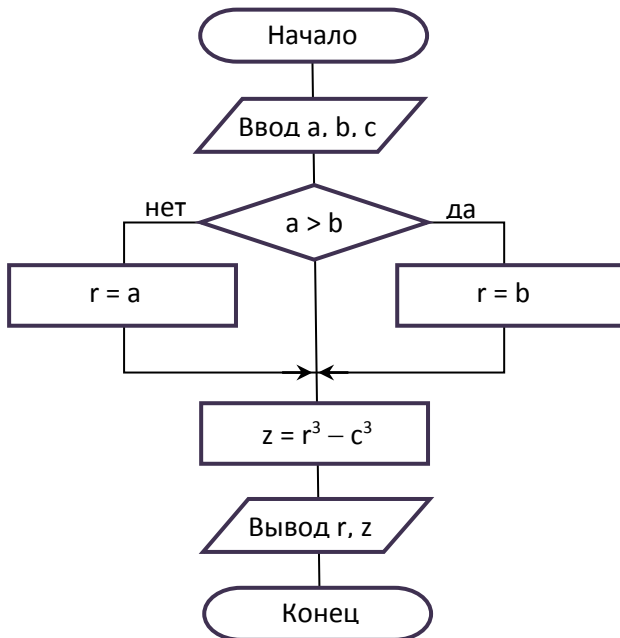
Шаг 3. Вычислить значение  $p = t^2 + z^2$  и перейти к шагу 5.

Шаг 4. Вычислить значение  $p = t^2 - z^2$ .

Шаг 5. Напечатать результат  $p$ .

Шаг 6. Прекратить вычисления.

2. Опишите словесно алгоритм, приведенный на рисунке:



3. Составьте алгоритм вычисления приближенного значения константы  $\pi$ , используя формулу:

$$\frac{\pi^2}{8} = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \dots = \sum_{n=1}^{\infty} \frac{1}{(2n-1)^2}$$

с погрешностью  $10^{-4}$ .

4. Предприниматель, начав дело, взял кредит размером  $S$  рублей под  $p$  процентов годовых и вложил в свое дело. По прогнозам, его дело должно давать прибыль  $R$  рублей в год. Сможет ли он накопить сумму, достаточную для погашения кредита, и если да, то через сколько лет?
5. Система  $N$  материальных точек на плоскости задана с помощью чисел  $x_1, y_1, m_1, \dots, x_N, y_N, m_N$ , где  $x_i, y_i$  – координаты  $i$ -ой точки, а  $m_i$  – ее масса,  $i = 1, 2, \dots, N$ . Постройте алгоритм вычисления координат центра масс системы, а также расстояния от центра масс до каждой из точек системы.

Указание. Координаты центра масс  $x_c, y_c$  могут быть вычислены по формулам:

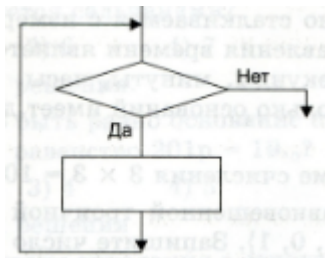
$$x_c = \frac{\sum_{i=1}^N m_i x_i}{\sum_{i=1}^N m_i}, \quad y_c = \frac{\sum_{i=1}^N m_i y_i}{\sum_{i=1}^N m_i}.$$

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение алгоритма.
2. Можно ли считать алгоритмом: (а) правила правописания; (b) законы физики; (с) математические формулы; (d) статьи уголовного кодекса. Ответы обоснуйте.
3. В повседневной жизни существует множество синонимов понятия «алгоритм». Что из перечисленного ниже нельзя назвать алгоритмом? Ответ обоснуйте.
  - 1) Рецепт приготовления блюда.
  - 2) Инструкцию по использованию бытового прибора.
  - 3) Афишу кинотеатра.
  - 4) План создания презентации, предлагаемый Мастером авто-содержания.
4. Какими способами можно описать алгоритм решения задачи?
5. Какую форму записи алгоритма называют блок-схемой? Приведите пример.
6. В чем суть таких свойств алгоритма как «результативность» и «массовость»? Обладает ли требованиям массовости и результативности следующая последовательность действий при вычислении значения функции  $y = (a + b)/c$ :
  - Шаг 1. Ввести значения переменных  $a$ ,  $b$  и  $c$ .
  - Шаг 2. Вычислить значение функции  $y = (a + b)/c$ .
  - Шаг 3. Напечатать значение результата  $y$ .
  - Шаг 4. Прекратить вычисления.
7. Какой алгоритм называют линейным? Приведите пример.
8. Какой алгоритм называется циклическим?

## Алгоритм и его свойства. Способы записи алгоритмов

9. К какому виду алгоритмических конструкций можно отнести фрагмент алгоритма, представленный на схеме?



- 1) Разветвляющийся с полным ветвлением.
- 2) Разветвляющийся с неполным ветвлением.
- 3) Цикл с предусловием.
- 4) Цикл с постусловием.

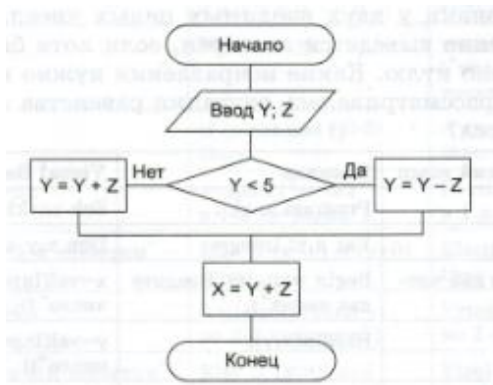
10. К какому виду циклических конструкций можно отнести фрагмент алгоритма, представленный на схеме?



- 1) Цикл с известным числом повторений.
- 2) Цикл с предусловием.
- 3) Цикл с постусловием.

11. На рисунке представлена блок-схема алгоритма. Какое значение будет иметь переменная X после выполнения алгоритма при начальном значении переменных  $Y = 5$ ,  $Z = -3$ ?

## Алгоритм и его свойства. Способы записи алгоритмов



12. К какому результату приведет исполнение представленного в виде блок-схемы алгоритма?

