

Составление алгоритмов для решения задач

ЧТО ТАКОЕ АЛГОРИТМ?



АЛГОРИТМ- это конечная последовательность шагов в решении задачи, приводящая от исходных данных к требуемому результату



Свойства алгоритма



Способы описания алгоритмов

- ❖ словесное описание;
- ❖ описание алгоритма с помощью математических формул;
- ❖ графическое описание алгоритма в виде блок-схемы;
- ❖ описание алгоритма с помощью псевдокода — программный;
- ❖ комбинированный способ.

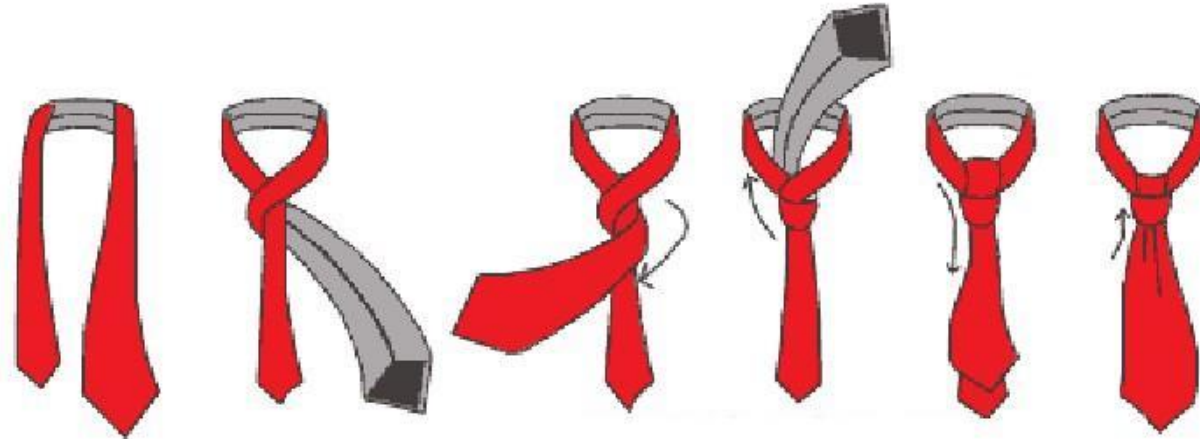
Словесное описание алгоритма

Словесное описание алгоритма представляет собой описание структуры алгоритма на естественном языке.

Достоинство: простота описания.

Недостаток: многословен и не имеет строгой формализации.

Примеры:



Описание алгоритма с помощью математических формул

$$Q = \frac{\pi}{4} \int_0^l d^2 dl.$$

$$\operatorname{ctg} 2\alpha = \frac{1 - \operatorname{tg}^2 \alpha}{2 \operatorname{tg} \alpha}$$

$$a^2 + b^2 = c^2$$

$$e^{i\pi} = -1$$

Описание алгоритма с помощью псевдокода – программный

Псевдокод основан на записи алгоритмов в формализованном представлении предписаний (специальная программа).

Достоинство: близость к языкам программирования.

Недостаток: сложность освоения и невозможность непосредственного ввода алгоритма для решения на ЭВМ.

Пример:

```
FLOYD-WARSHALL(M)
D0 = M           // M - матрица смежности
n = rows[M]      // размер матрицы

for k=1 to n do
  for i=1 to n do
    for j=1 to n do
      dkij = min(dk-1ij, dk-1ik + dk-1kj)
return Dn
```

```
ввод a, b, c
D = b2 - 4ac
если D < 0 то
  вывод Корней нет
иначе
  если D > 0 то
    x1 = ...
    x2 = ...
    вывод x1, x2
  иначе
    x1 = ...
    вывод x1
  конец если
```

Графическое описание алгоритма в виде блок-схемы

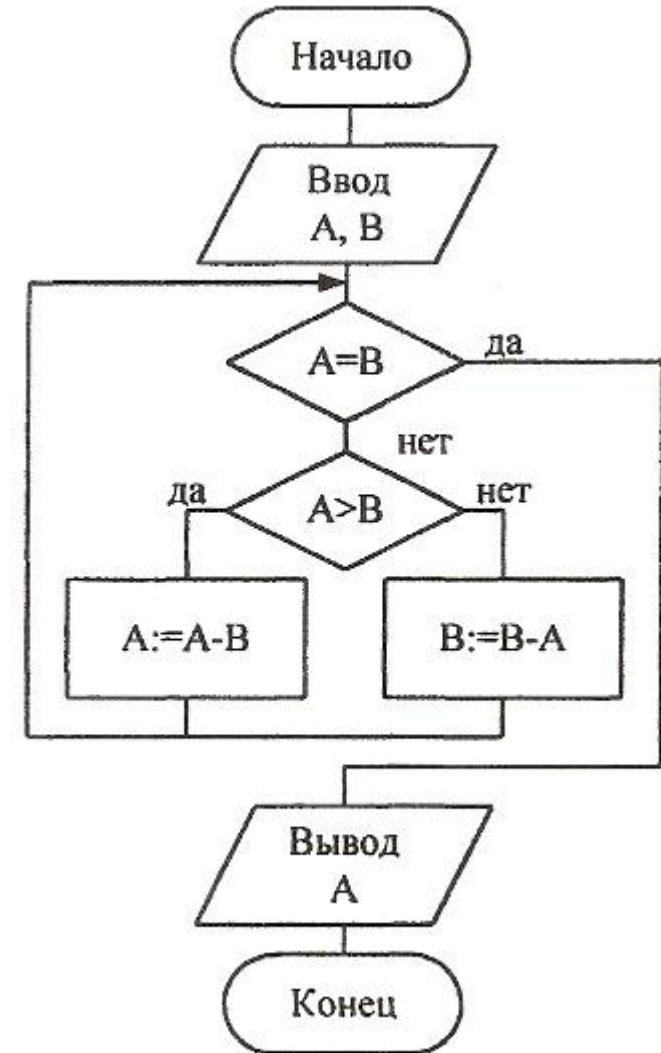
Графический способ предполагает, что для описания структуры алгоритма используется совокупность графических изображений (блоков), соединяемых линиями передачи управления.

Примеры:



Блок-схема

Блок-схема алгоритма – это графическое представление хода решения задачи.



Название блока	Обозначение	Назначение блока
Терминатор		Начало, завершение программы или подпрограммы
Процесс		Обработка данных (вычисления, пересылки и т. п.)
Данные		Операции ввода-вывода
Решение		Ветвления, выбор, итерационные и поисковые циклы
Подготовка		Счетные циклы
Граница цикла		Любые циклы
Предопределенный процесс		Вызов процедур
Соединитель		Маркировка разрывов линий
Комментарий		Пояснения к операциям



Виды алгоритмов

Линейный

Циклический

Разветвляющийся



Линейный алгоритм

Линейный – алгоритм, в котором команды выполняются однократно, одна за другой в том порядке, в котором записаны в программе.



Пример

Разработать алгоритм вычисления гипотенузы прямоугольного треугольника по известным значениям длин его катетов a и b .

Этап 1. Математическое описание решения задачи.

Математическим решением задачи является известная формула:

$$c = \sqrt{a * a + b * b}$$

где c -длина гипотенузы, a , b – длины катетов.

Этап 2. Определение входных и выходных данных.

Входными данными являются значения катетов a и b . Выходными данными является длина гипотенузы — c .

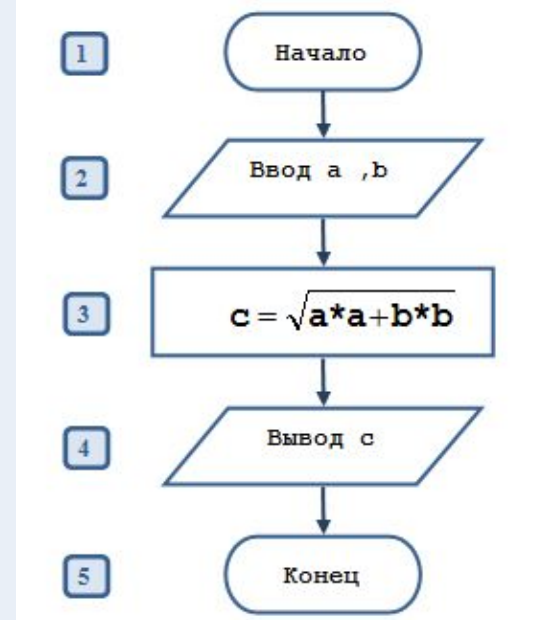
Этап 3. Разработка алгоритма решения задачи.

Словесное описание алгоритма

Запись алгоритма на языке блок-схем

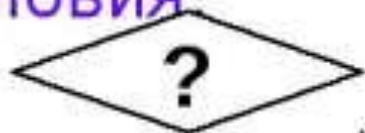
1. Начало алгоритма.
2. Ввод значений длин катетов a и b .
3. Вычисление длины гипотенузы c по формуле $c = \sqrt{a^2 + b^2}$
4. Вывод значения длины гипотенузы.
5. Конец алгоритма

На данной схеме цифрами указаны номера элементов алгоритма, которые соответствуют номерам пунктов словесного описания алгоритма.



Разветвляющийся алгоритм

Разветвляющийся – алгоритм, в котором последовательность действий зависит от результата выполнения условия



В разветвляющемся алгоритме имеются *разные варианты* решения задачи в зависимости от результата проверки какого-либо условия (*расходятся в разные стороны как веточки на дереве, отсюда название – разветвленный*).

Например, алгоритм проведения выходного дня в зависимости от погоды.

Если будет дождь – одни действия, если – нет, то планы будут другие.

Пример

Разработать алгоритм вычисления наибольшего числа из двух чисел x и y .

Этап 1. Математическое описание решения задачи.

Из курса математики известно, если $x > y$, то наибольшее число x , если $x < y$, то наибольшее число y , если $x = y$, то число x равно числу y .

Этап 2. Определение входных и выходных данных.

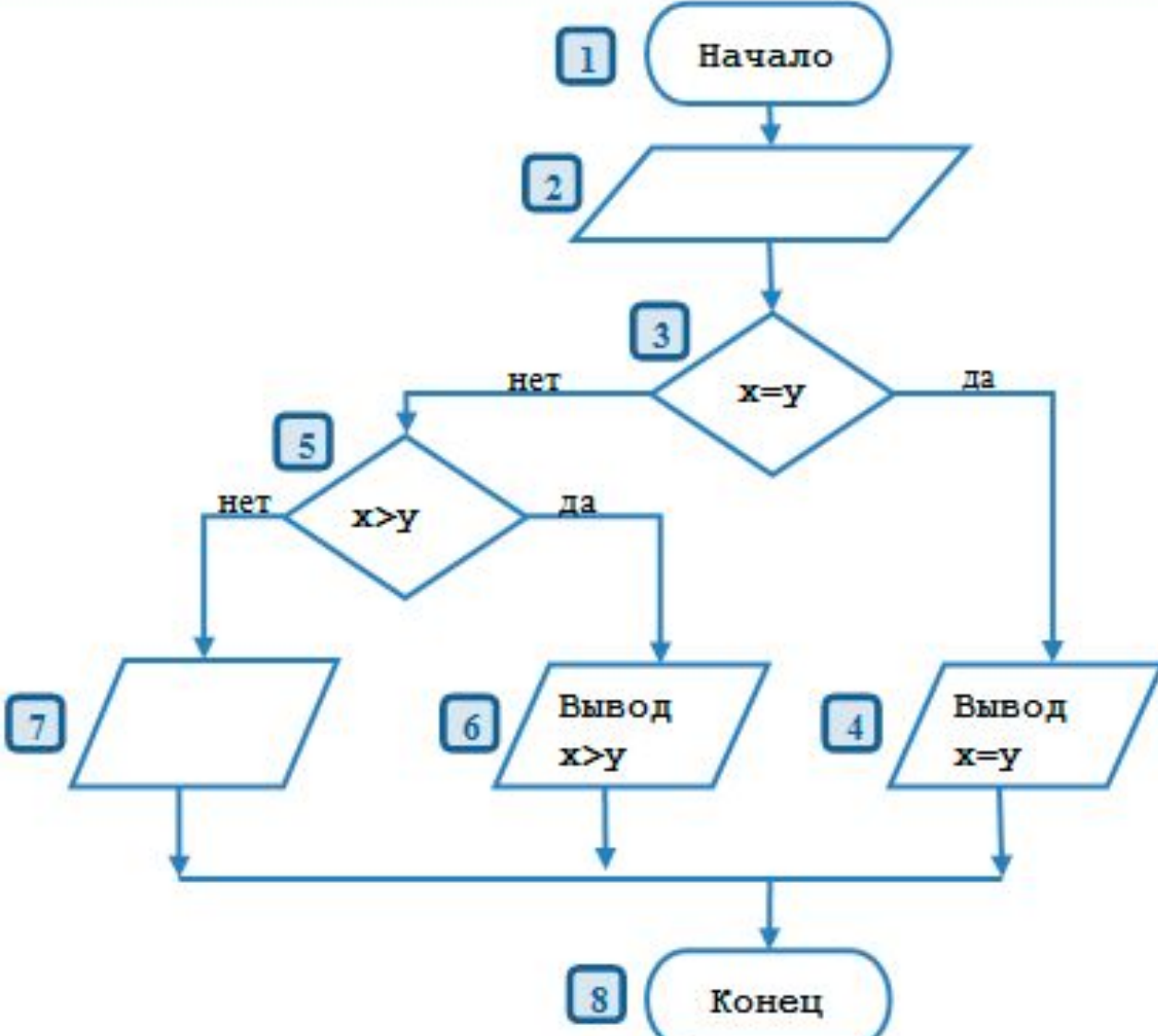
Входными данными являются значения чисел x и y .

Выходными данными являются:

- наибольшее число
- любое из чисел, если числа равны

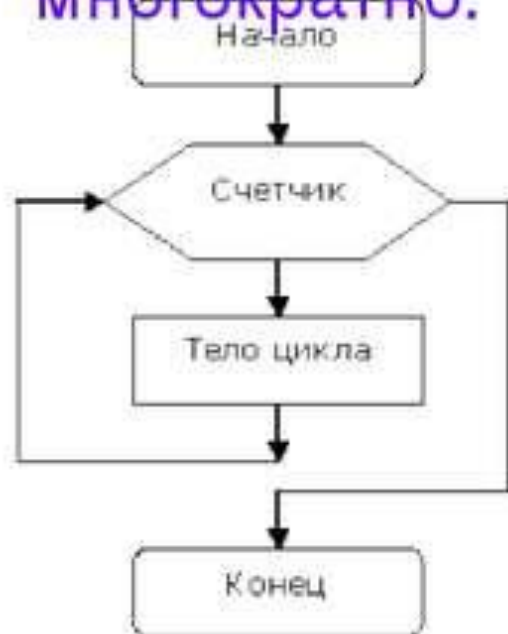
Для решения задачи нам необходимо знать значения x и y .

Этап 3. Разработка алгоритма решения задачи.

Словесное описание алгоритма	Запись алгоритма на языке блок-схем
<ol style="list-style-type: none">1. Начало алгоритма.2. Ввод значений x и y.3. Сравниваем x и y. Если $x = y$, то переход к шагу 4, иначе к шагу 5.4. Вывод информации: числа x и y равны. Переход к шагу 8.5. Сравниваем x и y. Если $x > y$, то переход к шагу 6, иначе к шагу 7.6. Вывод информации: число x больше y. Переход к шагу 8.7. Вывод информации: число y больше x. Переход к шагу 8.8. Конец алгоритма.	 <pre>graph TD; 1([1 Начало]) --> 2[/2 /]; 2 --> 3{x=у}; 3 -- да --> 4[/4 Вывод x=y /]; 3 -- нет --> 5{x>y}; 5 -- да --> 6[/6 Вывод x>y /]; 5 -- нет --> 7[/7 Вывод y>x /]; 4 --> 8([8 Конец]); 6 --> 8; 7 --> 8;</pre>

Циклический алгоритм

Циклический – алгоритм, в котором определенный набор команд выполняется многократно.



- это описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие.



Циклическая структура

- параметр цикла;
- начальное и конечное значения параметров цикла;
- шаг цикла.

Виды циклических алгоритмов

Цикл с
предусловием



Цикл типа *Пока*

Цикл с
постусловием



Цикл типа *ДО*

Цикл с
параметром



Цикл типа *ДЛЯ*



Пример

Разработать алгоритм вычисления суммы натуральных чисел от 1 до 100.

Этап 1. Математическое описание решения задачи.

Обозначим сумму натуральных чисел через S . Тогда формула вычисления суммы натуральных чисел от 1 до 100 может быть записана так:

$$S = 1 + 2 + 3 + \dots + 97 + 98 + 99 + 100 = \sum_{i=1}^n X_i$$

где X_i – натуральное число X с номером i , который изменяется от 1 до n , $n=100$ – количество натуральных чисел.

Этап 2. Определение входных и выходных данных.

Входными данными являются натуральные числа: 1, 2, 3, 4, 5, ..., 98, 99, 100.

Выходные данные – значение суммы членов последовательности натуральных чисел.

Параметр цикла – величина, определяющая количество повторений цикла. В нашем случае i – номер натурального числа.

Подготовка цикла заключается в задании начального и конечного значений параметра цикла.

начальное значение параметра цикла равно 1,

конечное значение параметра цикла равно n ,

шаг цикла равен 1.

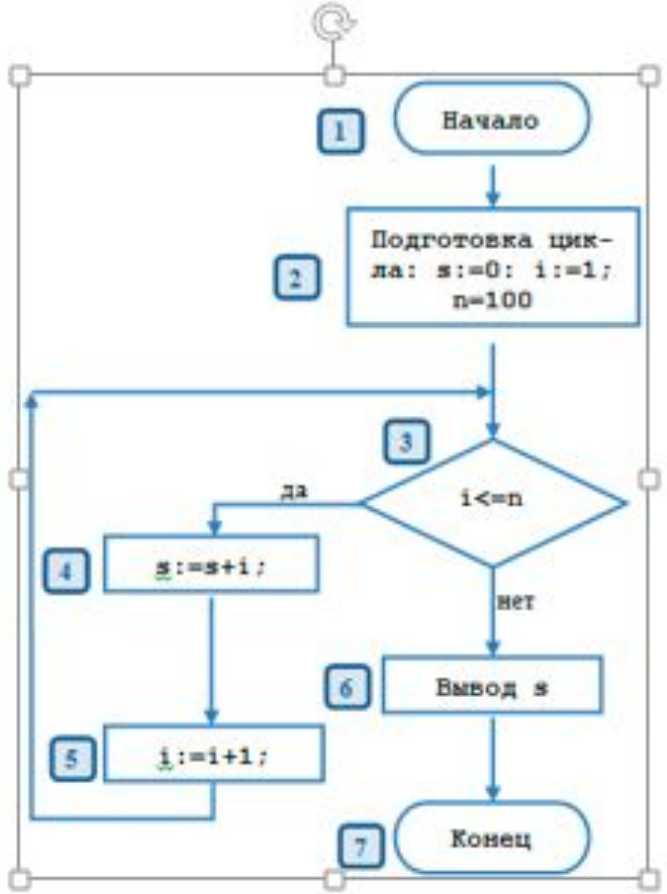
Для корректного суммирования необходимо предварительно задать начальное значение суммы, равное 0.

Тело цикла. В теле цикла будет выполняться накопление значения суммы чисел, а также вычисляться следующее значение параметра цикла по формулам:

$$S=S+i; \quad I=I+1;$$

Условие продолжения цикла: цикл должен повторяться до тех пор, пока не будет добавлен последний член последовательности натуральных чисел, т.е. пока параметр цикла будет меньше или равен конечному значению параметра цикла.

Этап 3. Разработка алгоритма решения задачи.

Словесное описание алгоритма	Запись алгоритма на языке блок-схем
<ol style="list-style-type: none">1. Начало алгоритма.2. Подготовка цикла: $S:=0; i:=1; n:=100;$3. Проверка условия. Если $i \leq n$, то перейти к шагу 4, иначе к шагу 6.4. Накопление суммы: $S:=S+i;$5. Вычисление следующего значения параметра цикла: $i:=i+1;$6. Вывод информации: сумма натуральных чисел – S.7. Конец алгоритма.	<p>В схеме алгоритма решения задачи цифрами указаны номера элементов алгоритма. Номера элементов соответствуют номерам шагов словесного описания алгоритма.</p>  <pre>graph TD; Start([Начало]) --> Init[Подготовка цикла: s:=0; i:=1; n:=100]; Init --> Cond{i <= n}; Cond -- да --> Sum[s:=s+i;]; Sum --> Inc[i:=i+1;]; Inc --> Cond; Cond -- нет --> Output[Вывод s]; Output --> End([Конец]);</pre>

Задача 1

Алгоритм вычисления значения выражения $K=3b+6a$.

Ввод исх. данных: b, a

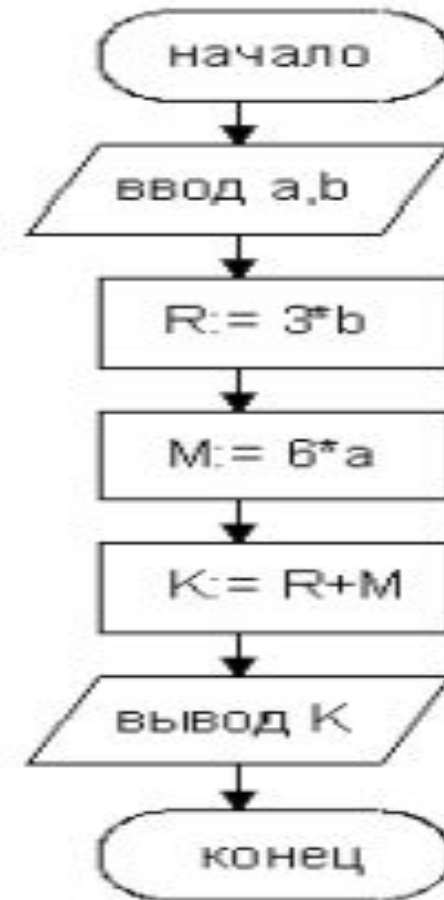
Вычисление $R := 3*b$

Вычисление $M := 6*a$

Вычисление $K := R+M$

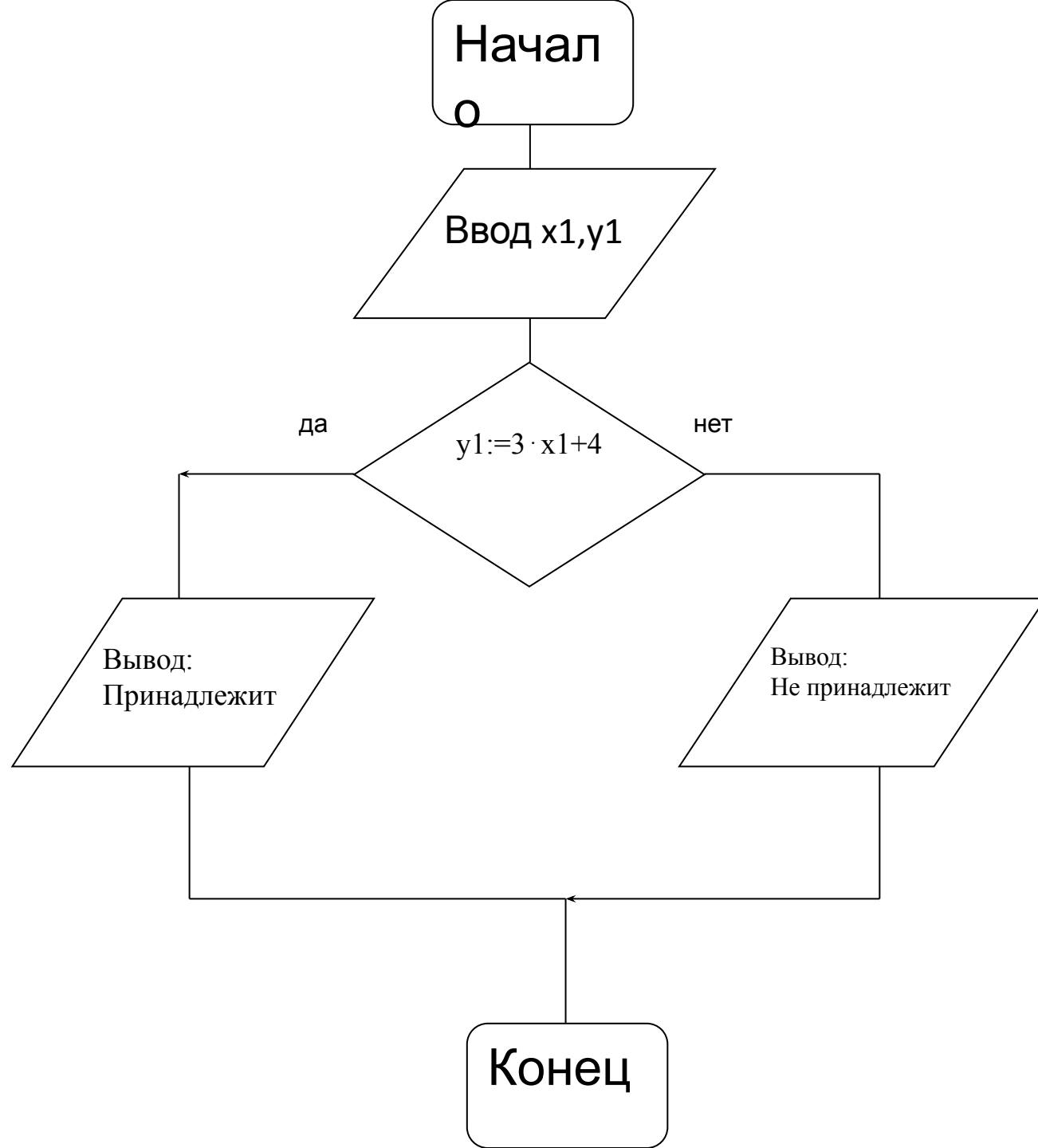
Вычисление $K=3*b+6*a$

Вывод результата: K



Задача 2

Алгоритм, определяющий, пройдет ли график функции $y=3x+4$ через точку с координатами x_1, y_1 .



Задача 3

Алгоритм, определяющий факториал натурального числа n :

$$n! = 1 * 2 * 3 * \dots * (n-1) * n$$

$$0! = 1$$

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

