Исполнитель Калькулятор

Алгоритмы

Алгоритм – это четко определенный план действий для исполнителя.

Свойства алгоритма

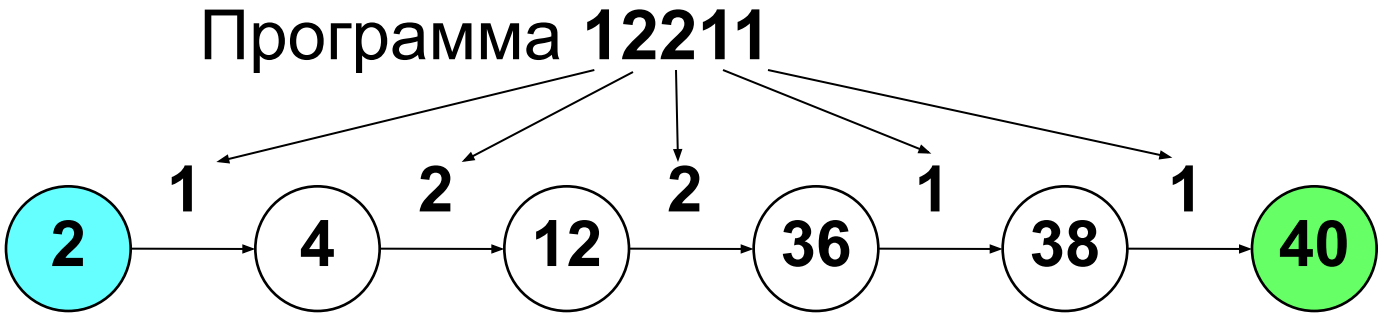
- дискретность: состоит из отдельных шагов (команд)
- понятность: должен включать только команды, известные исполнителю (входящие в СКИ)
- определенность: при одинаковых исходных данных всегда выдает один и тот же результат
- конечность: заканчивается за конечное число шагов
- массовость: может применяться многократно при различных исходных данных
- корректность: дает верное решение при любых допустимых исходных данных

Система команд

Исполнитель Калькулятор работает с одним числом и умеет выполнять с ним две операции (команды):

1. **прибавь 2**
2. **умножь на 3**

Программа – это последовательность номеров команд, которые нужно выполнить.



начальное число

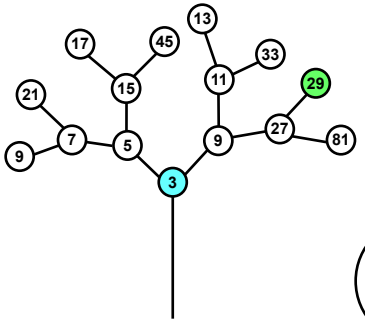
результат

Обратная задача (составление программы)

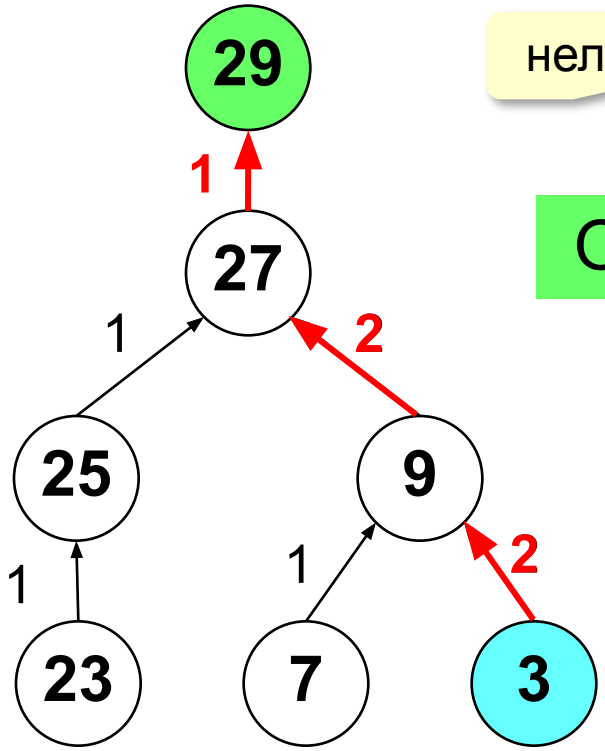
Используя команды:

1. **прибавь 2**
2. **умножь на 3**

написать программу, которая из 3 получает 29.

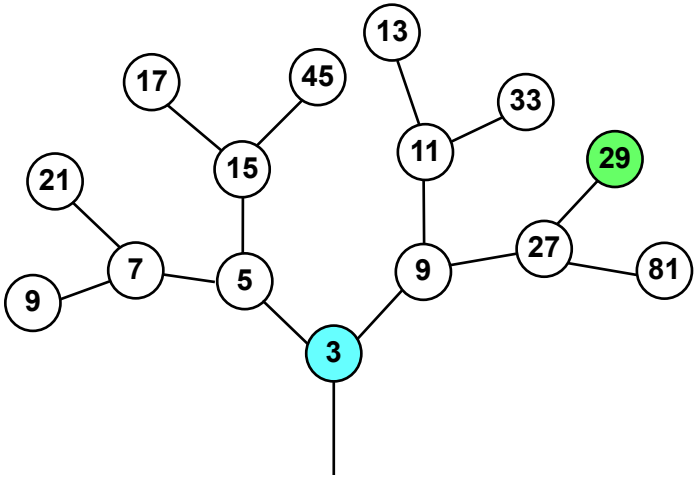


Обратная задача (решение «с конца»)



нельзя делить на 3!

Ответ: **221**



? Почему решение «с конца» короче?

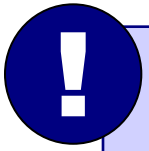
! Решение «с конца» короче, если в списке команд есть **необратимая операция** (каждое целое число можно умножить на 3, но не каждое делится на 3)!

Ещё пример

Используя команды:

1. **прибавь 2**
2. **умножь на 3**

написать программу, которая из 2 получает 15.



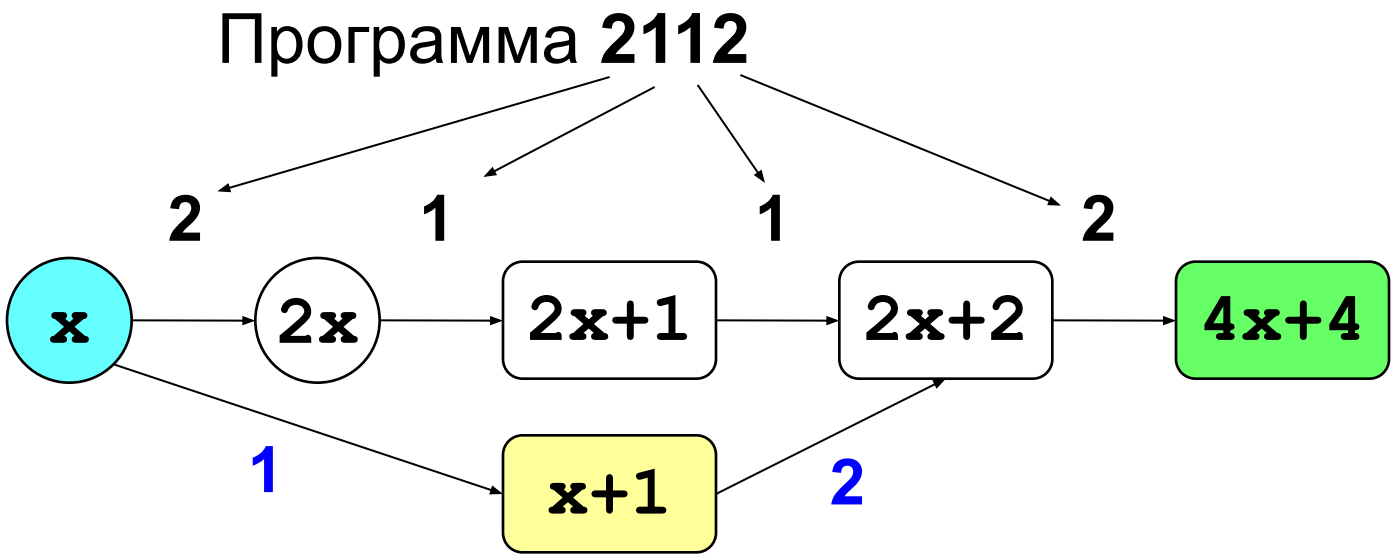
Не все задачи этого типа решаемы. Разрешимость зависит от системы команд и начального числа.

Удвоитель

У исполнителя есть команды:

1. **прибавь 1**
2. **умножь на 2**

Дана программа: 2112. Как можно сделать то же самое за 3 шага?



Удвоитель

У исполнителя есть команды:

1. **прибавь 1**
2. **умножь на 2**

Задания:

1) Какие числа можно получить из 0?
2) Как из числа 5 получить 105?
3) Какие числа можно получить из отрицательного числа N?
4) Как построить самую короткую программу для получения заданного **x** из **0**?
5) Найдите минимальное число, которое может быть получено из 0 только за 6 шагов.

Удвоитель

У исполнителя есть команды:

1. **прибавь 1**
2. **умножь на 2**

Докажите, что:

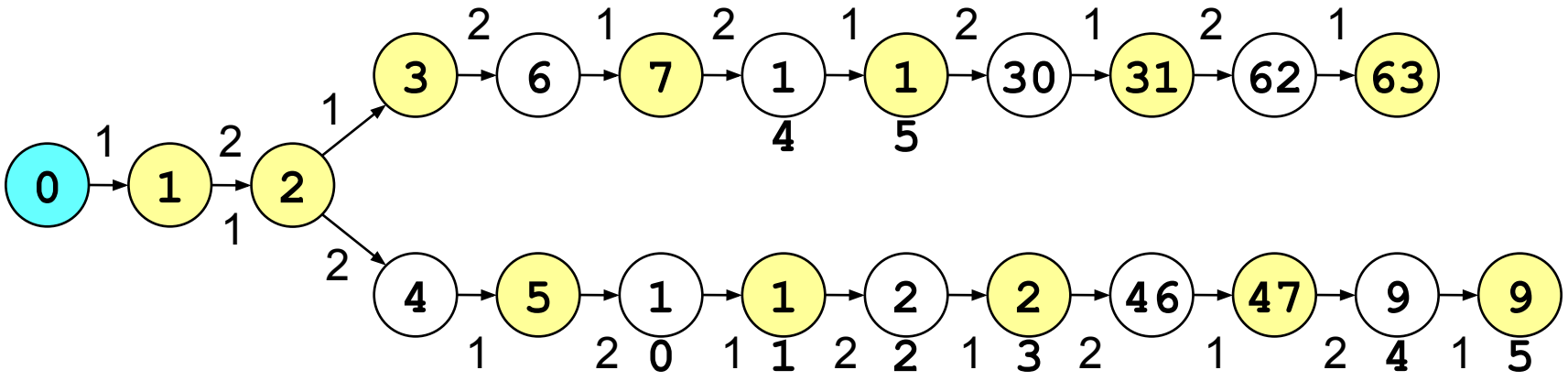
1) любое число, меньшее 10, можно получить из 0 за 5 шагов

2) любое число, меньшее 100, можно получить из 0 за 12 шагов

Длина оптимальной программы

Минимальное число, для которого *оптимальная* программа содержит ровно *N* команд:

- первая команда – 1 (0 →1)
- программа оканчивается на 1 (**прибавь 1**)
- при «обратном ходе» команды 1 и 2 чередуются



Раздвоитель

У исполнителя есть команды:

1. **вычти 1**
2. **раздели на 2**

Задания:

1) Какие числа можно получить из положительного числа N?

2) Какие числа можно получить из отрицательного числа N?

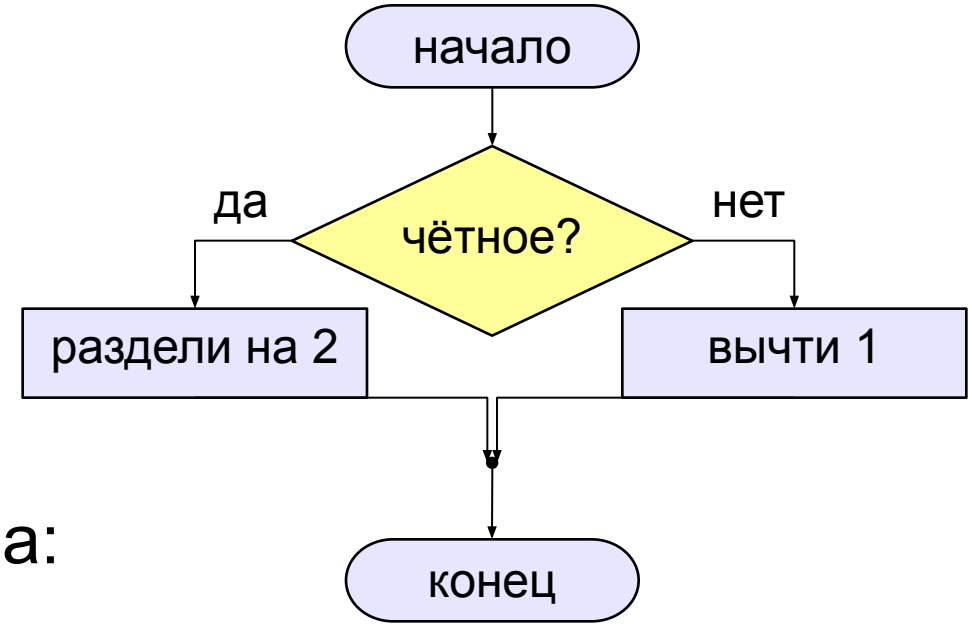
3) Как быстрее всего получить 0 из положительного числа N?

Раздвоитель (ветвление)

Алгоритм:

```
если четное
   то раздели на 2
   иначе вычти 1
все
```

Блок-схема:



Что получится для числа:

35 → 34
44 → 22
77 → 76
88 → 44

Раздвоитель (циклы)

Цикл – это повторение одинаковых действий.

Алгоритм:

начало цикла

тело цикла

```
нц 5 раз
   если четное
      то раздели на 2
      иначе вычти 1
   всё
кц
```

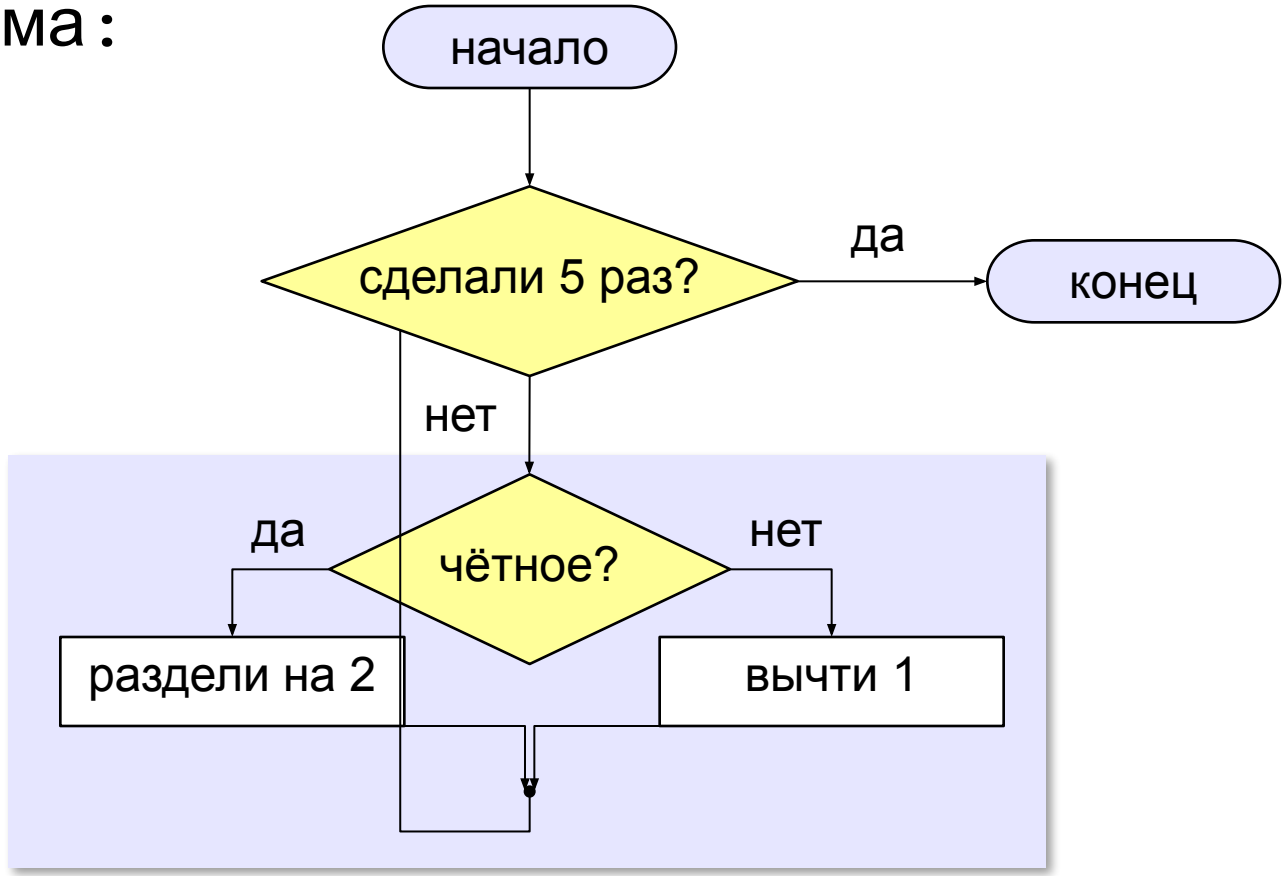
конец цикла

Что получится:

10 → 0
20 → 1
30 → 3
50 → 3
60 → 6

Раздвоитель (циклы)

Блок-схема:

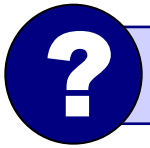


тело цикла

Раздвоитель (циклы)

Алгоритм:

```
нц пока положительное
  если четное
    то раздели на 2
    иначе вычти 1
  всё
кц
```



Что получим?

Задание: нарисуйте блок-схему.

Сколько шагов цикла выполнится для числа

15 → 7

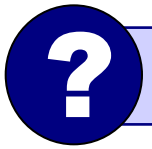
16 → 5

128 → 8

Раздвоитель (циклы)

Алгоритм получения 0 из положительного числа:

```
нц пока положительное
   нц пока четное
      раздели на 2
   кц
   вычти 1
кц
```



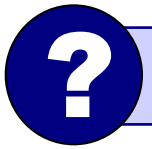
Всегда ли работает?

Задание: нарисуйте блок-схему.

Раздвоитель (циклы)

Алгоритм получения 0 из положительного числа:

```
нц пока положительное
   вычти 1
   нц пока четное
      раздели на 2
   кц
кц
```



Всегда ли работает?

Задание: нарисуйте блок-схему.

Раздвоитель (циклы)

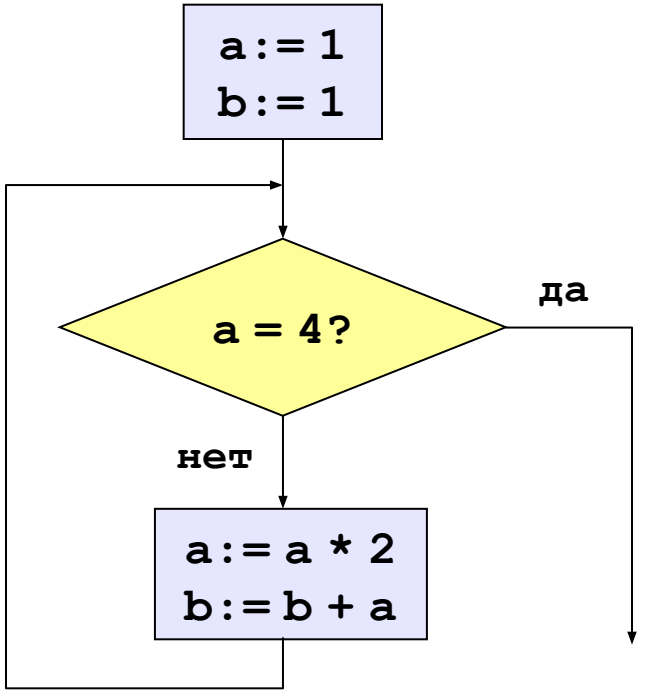
Алгоритм получения 0 из положительного числа:

```
нц пока положительное
  если нечетное
   то вычти 1
  всё
  нц пока четное
    раздели на 2
  кц
кц
```

? Всегда ли работает?

Задание: нарисуйте блок-схему.

Анализ блок-схем

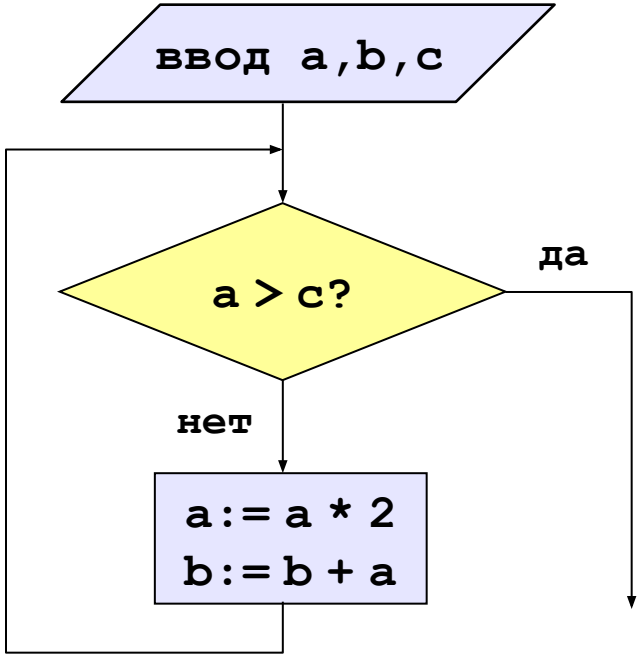


	a	b
	○	
		○

? Что будет при a = 3? a = 4? a = 5?

Анализ блок-схем

Напишите программу, в которой **a**, **b** и **c** вводятся с клавиатуры. Заполните таблицу:

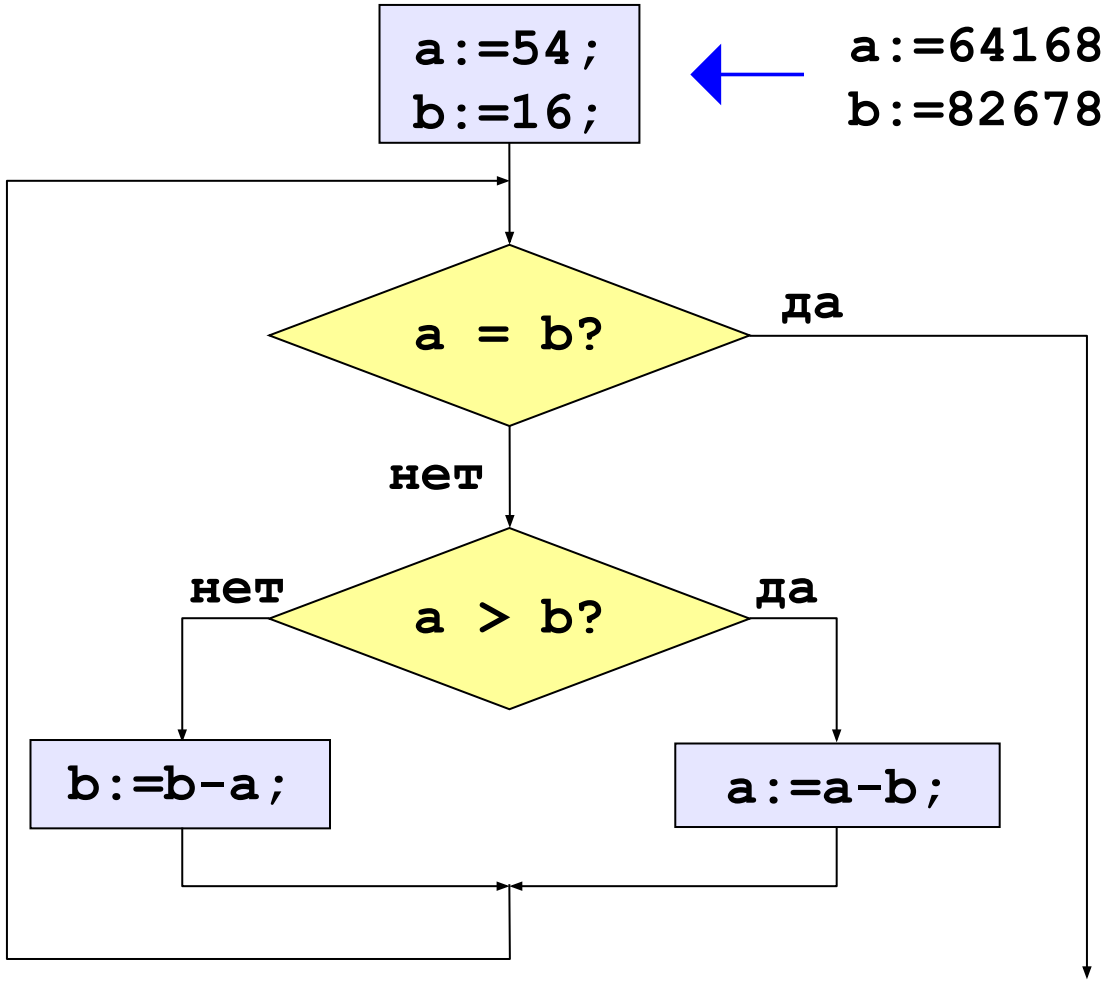


Исходные данные			Результат	
a	*b*	*c*	*a*	*b*
2	3	4		
5	12	100		
3	25	999		
111	222	9999		
111	222	111		
100	12	5		

? Как вывести результат?

```
вывод a, b      → 85
вывод a, " ", b → 8 5
вывод "a=", a, "b=", b
                → a=8 b=5
```

Анализ блок-схем



Напишите программу, в которой **a** и **b** вводятся с клавиатуры. Что она вычисляет?

Алгоритм Евклида

Надо: вычислить наибольший общий делитель (НОД) чисел **a** и **b**.

Заменяем большее из двух чисел разностью большего и меньшего до тех пор, пока они не станут равны. Это и есть НОД.

```
НОД(a,b)=НОД(a-b,b)
        =НОД(a,b-a)
```

Евклид
(365-300 до. н. э.)

Пример:

```
НОД (14, 21) = НОД (14, 21-14) = НОД (14, 7)
             = НОД (7, 7) = 7
```

много шагов при большой разнице чисел:

```
НОД (1998, 2) = НОД (1996, 2) = … = 2
```

Модифицированный алгоритм Евклида

Заменяем большее из двух чисел остатком от деления большего на меньшее до тех пор, пока меньшее не станет равно нулю. Тогда большее — это НОД.

```
НОД(a,b) = НОД(mod(a,b), b)
         = НОД(a, mod(b,a))
```

Пример:

```
НОД (14, 21) = НОД (14, 7) = НОД (0, 7) = 7
```

Алгоритм Евклида

Составить программу для вычисления НОД с помощью алгоритма Евклида и заполнить таблицу:

a	64168	358853	6365133	17905514	549868978
b	82678	691042	11494962	23108855	298294835
НОД(a,b)					

«5»: Подсчитать число шагов алгоритма.

a	64168	358853	6365133	17905514	549868978
b	82678	691042	11494962	23108855	298294835
НОД(a,b)					
шагов					

Конец фильма

ПОЛЯКОВ Константин Юрьевич
д.т.н., учитель информатики высшей категории,
ГОУ СОШ № 163, г. Санкт-Петербург
kpolyakov@mail.ru

Использованы материалы Д. Кириенко, школа № 179, г. Москва

