

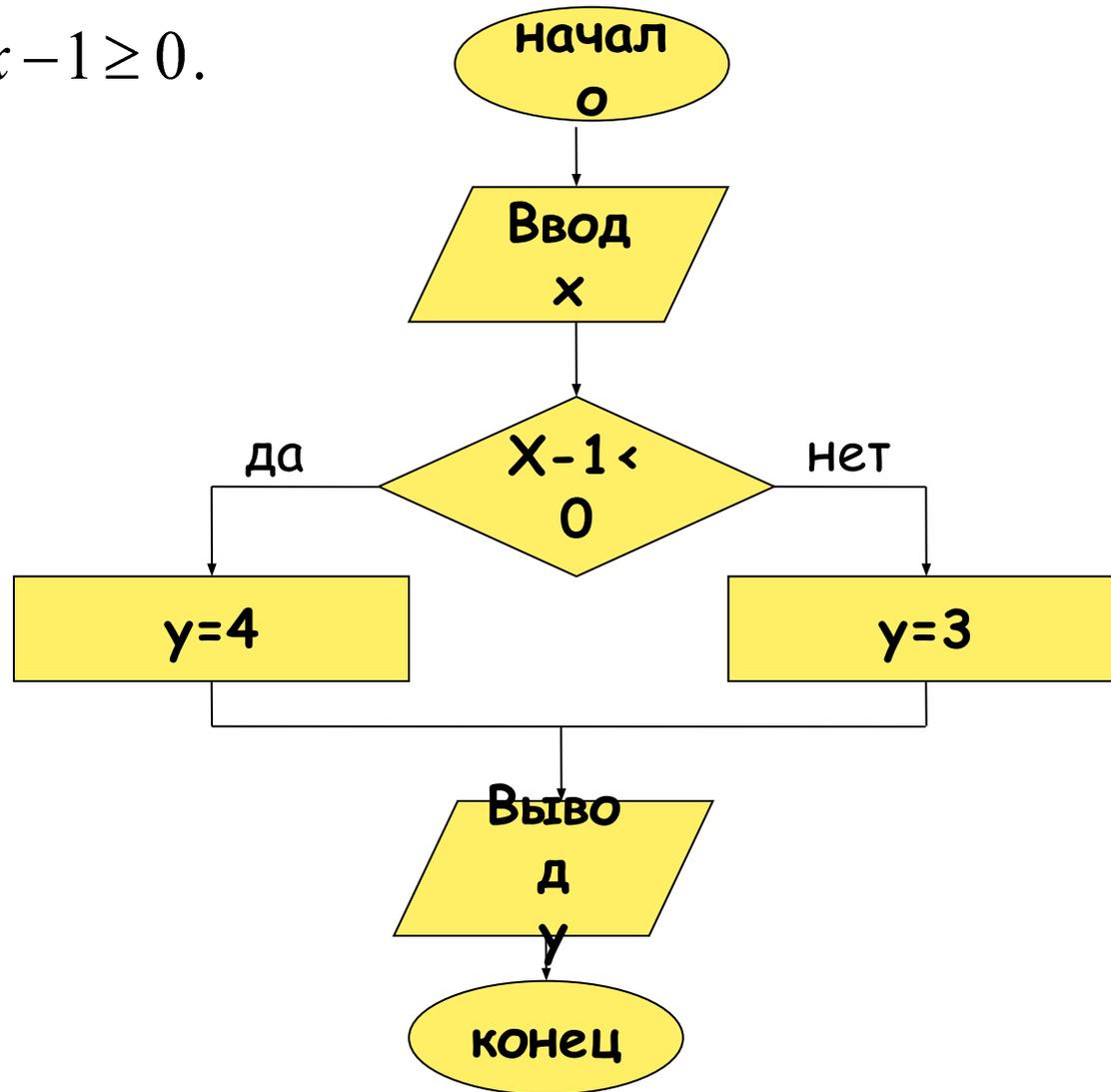
Разветвляющиеся алгоритмы на языке Паскаль

- **Разветвляющийся алгоритм**
- алгоритм, в котором в зависимости от выполнения или не выполнения некоторого условия совершается либо одна, либо другая последовательность действий.

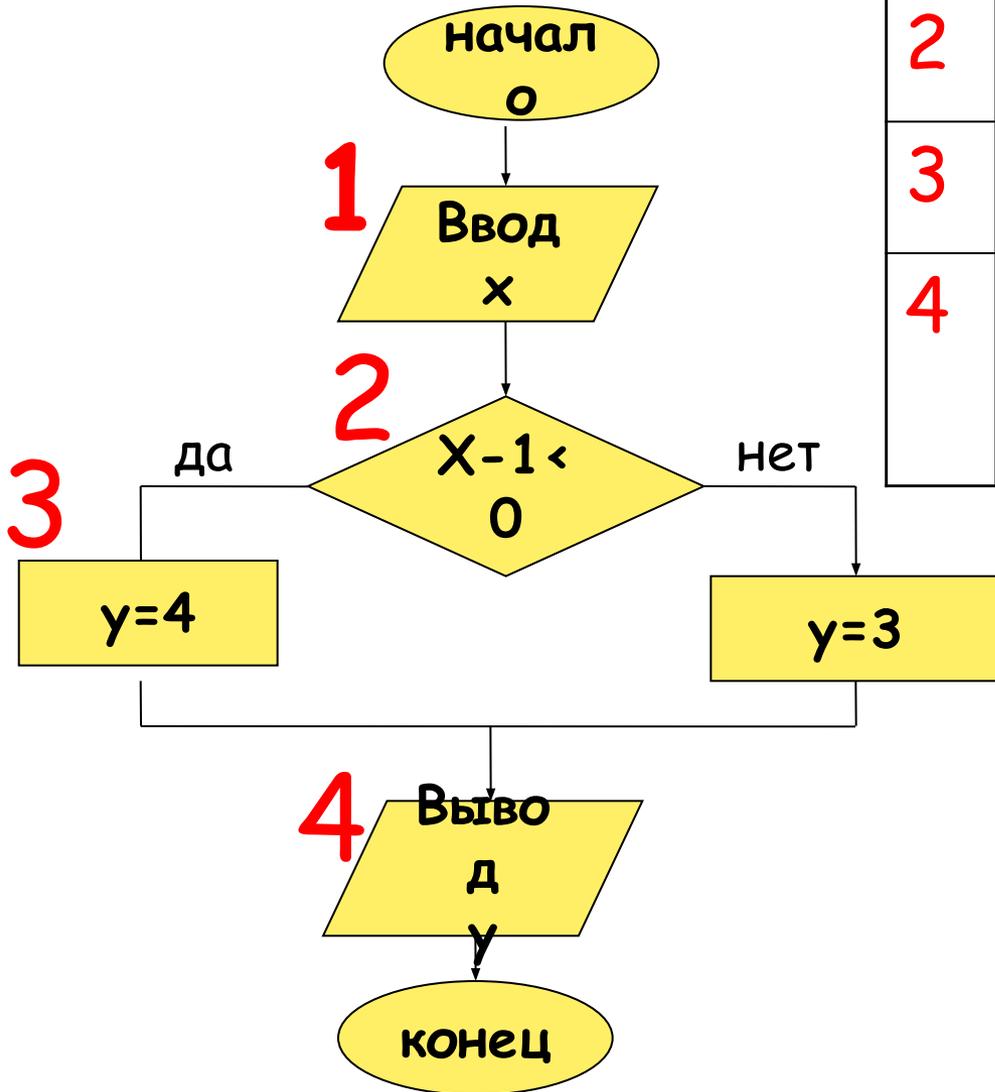
Задача 1: в блок - схеме алгоритма
вычисления значения функции

$$y = \begin{cases} 4, & \text{если } x-1 < 0; \\ 3, & \text{если } x-1 \geq 0. \end{cases}$$

заполните пустые блоки.

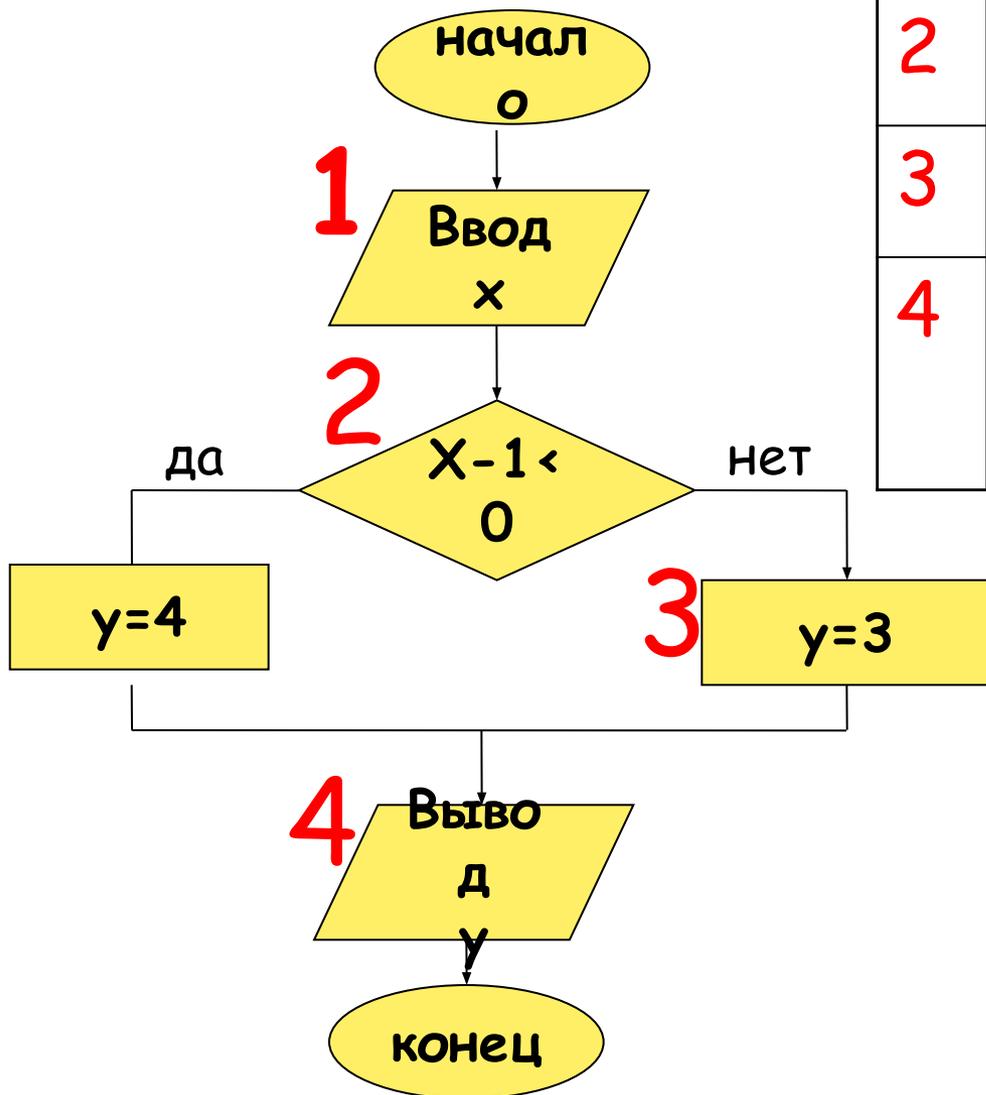


$X = -5$



1	Ввод x	-5
2	$X-1 < 0$	$-5-1 < 0$, да
3	y=4	
4	Вывод y	4

X=5



1	Ввод x	5
2	$X-1 < 0$	$5-1 < 0$, нет
3	$y=3$	
4	Вывод y	3

ПОЛНАЯ ФОРМА ВЕТВЛЕНИЯ

НА АЛГОРИТМИЧЕСКОМ
ЯЗЫКЕ

Если условие
то серия команд 1
иначе серия команд 2
конец ветвления

НА ЯЗЫКЕ ПАСКАЛЬ

IF <условие>
then <серия команд 1>
else <серия команд 2>;

НЕПОЛНАЯ ФОРМА ВЕТВЛЕНИЯ

НА АЛГОРИТМИЧЕСКОМ
ЯЗЫКЕ

Если условие
то серия команд 1
конец ветвления

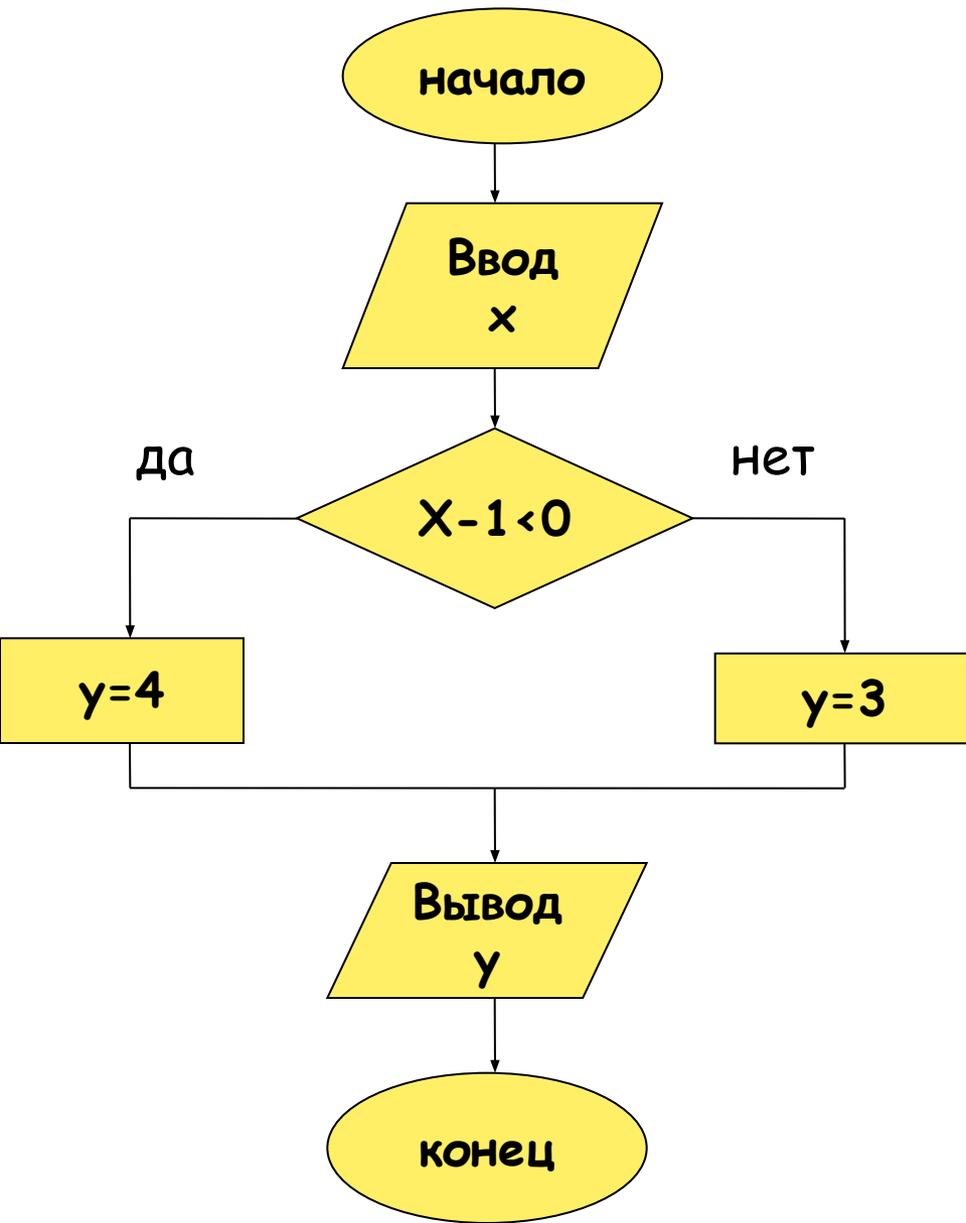
НА ЯЗЫКЕ ПАСКАЛЬ

IF <условие>
then <серия команд 1>;

Операции отношения:

- < - меньше
- > - больше
- <= - меньше или равно
- >= - больше или равно
- <> - не равно
- = - равно

Задание: по известной блок-схеме составить программу на языке Паскаль



```
Program z1;
```

```
Var   y: Integer;  
      x: Real;
```

```
Begin
```

```
  Write('vvedi x= ');
```

```
  Read ( x);
```

```
  If x-1 < 0 then
```

```
    y:=4
```

```
  else y:=3;
```

```
  Write ( ' y= ' , y);
```

```
  Readln;
```

```
End.
```

Поиск наименьшего значения из трех чисел x, y и z.

Решение состоит в следующем: сначала сравниваем два числа x и y (**полная форма**), а затем наименьшее из них (min) сравниваем с третьим числом z. Если z меньше минимального, то присваиваем минимальному значение z, иначе ничего не делаем (**неполная форма**).

```
VAR x, y, z, min: real;
```

```
BEGIN
```

```
READ(x, y, z);
```

```
if x < y then min := x else min := y;
```

```
if z < min then min := z;
```

```
Write(min);
```

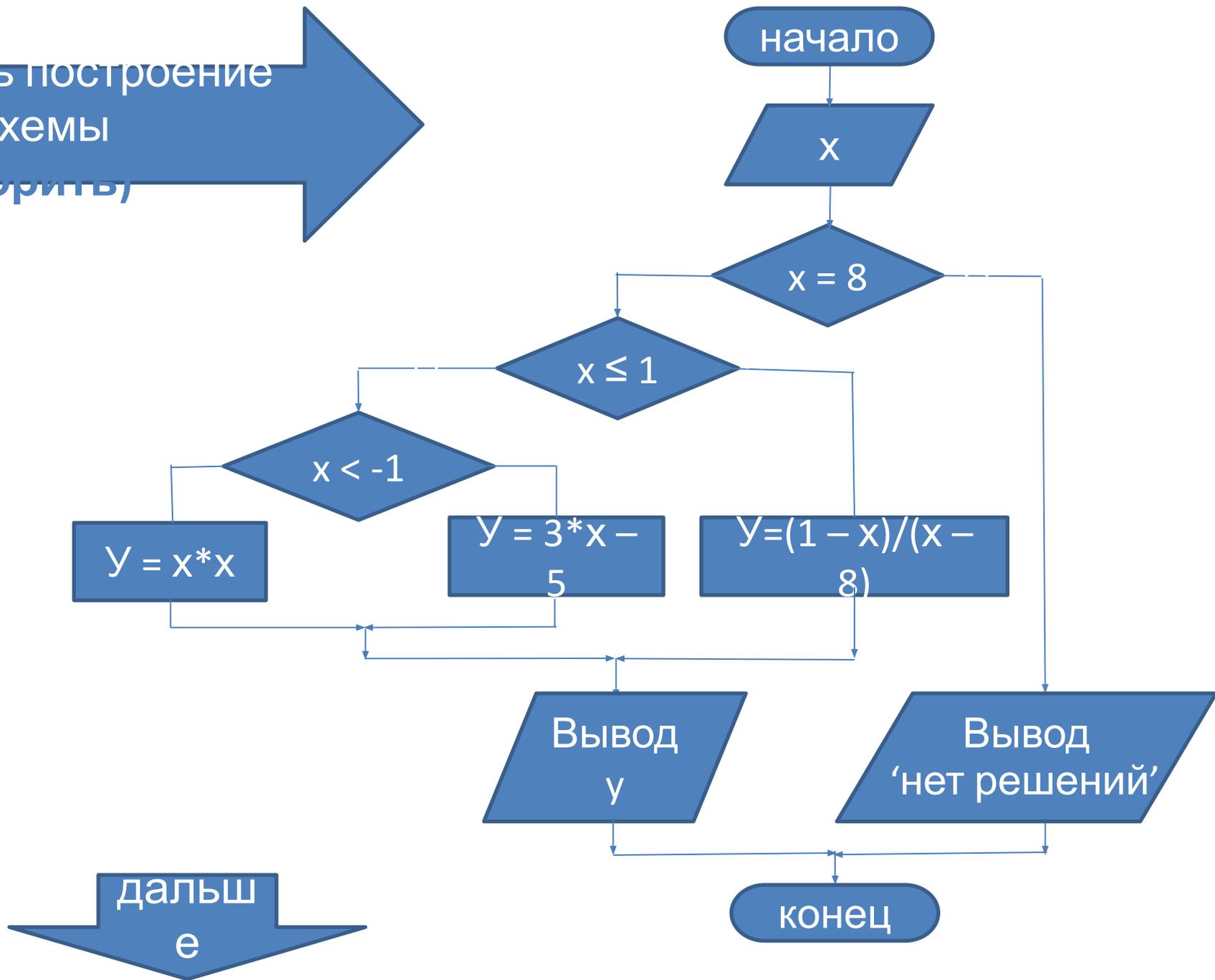
```
END
```

Вычисление значения функции, в зависимости от значения аргумента

$$y = \begin{cases} 1 \text{ формула} & x < a \\ 2 \text{ формула} & a \leq x \leq b \\ 3 \text{ формула} & \text{в остальных случаях} \end{cases}$$

В таких задачах **эффективней использовать вложенный условный оператор**, чем последовательную проверку каждого интервала. Используя вложенный условный оператор, мы уменьшаем **количество операторов до одного** вместо того количества, которое соответствует количеству возможных вариантов решения. Также уменьшается общее количество операций отношения в проверяемых условиях (ветвь **else** исключает ранее проверенное условие), что делает алгоритм **быстрым**.

Начать построение
блок-схемы
(повторить)



Перевод алгоритма на Pascal

```
Var x, y: real;  
BEGIN  
Read(x);  
if x=8  
  then WRITE('Нет решений')  
  else begin  
    if x<=1  
      then if x<-1  
        then y:=sqr(x)  
        else y:=3*x-5  
      else y:=(1-x)/(x-8);  
    write(y);  
  end;  
END.
```

Замена значения одной величины из трех заданных

В этом случае эффективней вместо трех последовательных условных операторов использовать один с вложением двух других.

Составить алгоритм, который наибольшее из трех вещественных чисел a , b , c заменяет их средним значением.

if ($a > b$) and ($a > c$) then $a := (a + b + c) / 3$

else

if ($b > a$) and ($b > c$) then $b := (a + b + c) / 3$

else

if ($c > b$) and ($c > a$) then $c := (a + b + c) / 3;$