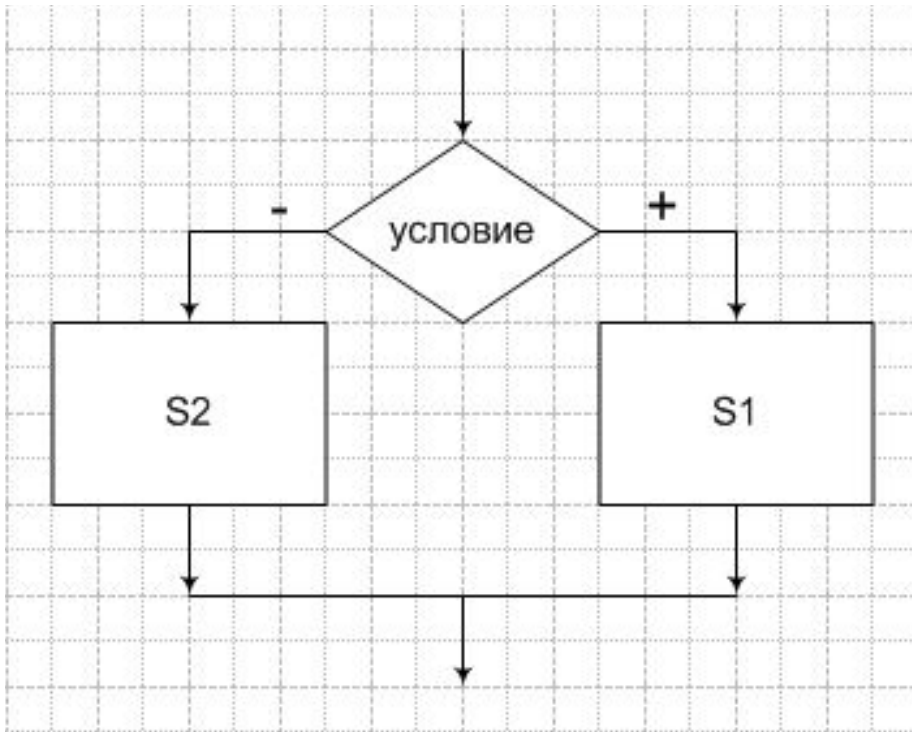


Разветвляющиеся алгоритмы и программы

1. Полная форма ветвлений

С помощью структурных блок-схем



Выполняется проверка условия

Если условие верно, то выполняются команды по правой ветке

Если условие не верно, то выполняются команды по левой ветке

1. Полная форма ветвлений

На языке Pascal

Формат:

```
if <условие > then <операторы 1>  
    else <операторы 2>;
```

(если... то...иначе ...)

Проверяется условие

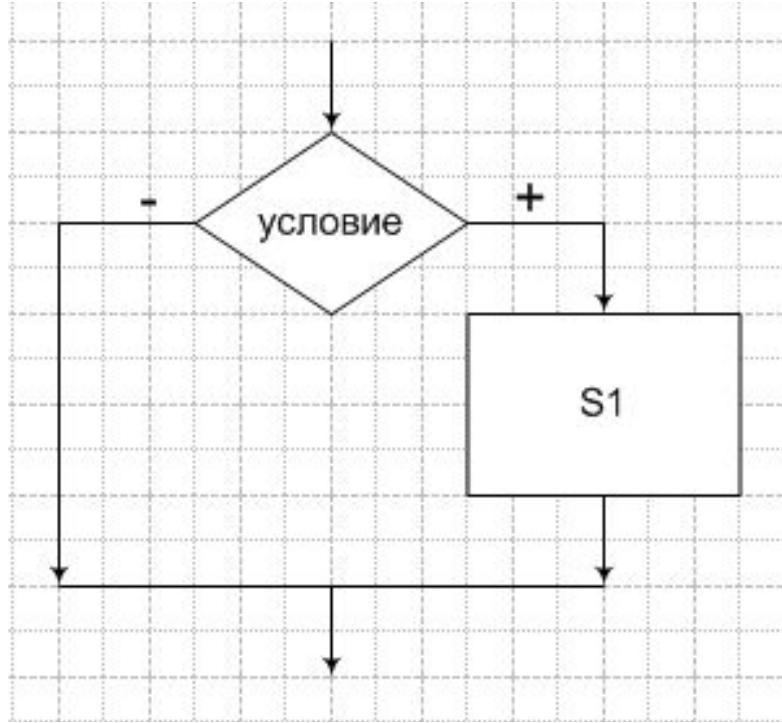
Если условие верно, то выполняются операторы после служебного слова **then**, затем выполняется переход на следующую строку программы

Если условие не верно, то выполняются операторы после слова **else**, затем происходит переход на следующую строку программы.

Полную форму ветвления наиболее удобно применять, когда в задачах выбор по двум вариантам.

2. Сокращенная форма ветвлений

С помощью структурных блок-схем



Выполняется проверка условия

Если условие верно, то выполняются команды по правой ветке

Если условие не верно, то по левой ветке происходит переход на следующие команды

2. Сокращенная форма ветвлений

На языке Pascal

Формат:

```
if <условие > then < операторы >;
```

(если... то...)

Проверяется истинность условия

Если условие верно, то выполняются операторы после служебного слова **then**, затем выполняется переход на следующую строку программы.

Если условие не верно, то выполнение передается на следующую строку программы сразу

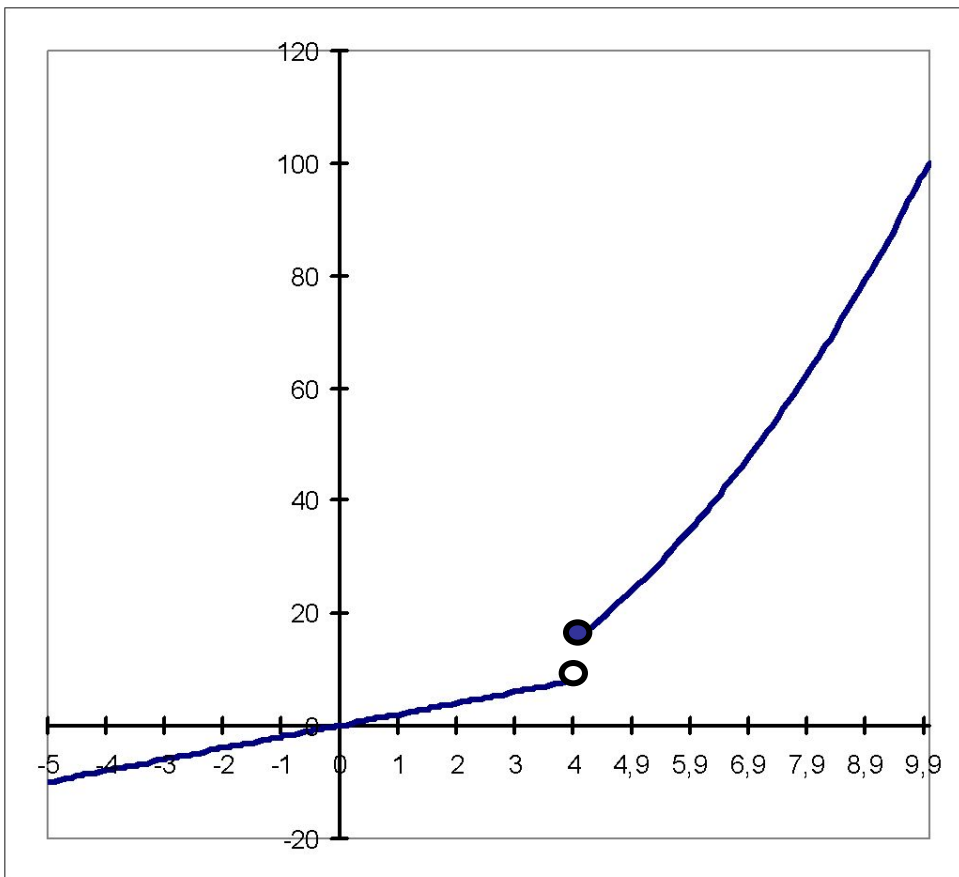
Пример

Задание.

Составить программу вычисления значения функции

$$y = \begin{cases} x^2, & x \geq 4 \\ 2x, & x < 4 \end{cases}$$

1. Математическая модель



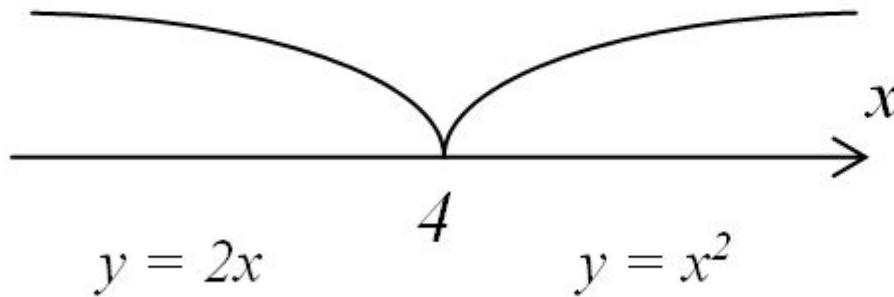
Пример 1

Задание.

Составить программу вычисления значения функции

$$y = \begin{cases} x^2, & x \geq 4 \\ 2x, & x < 4 \end{cases}$$

1. Математическая модель



Дано: x

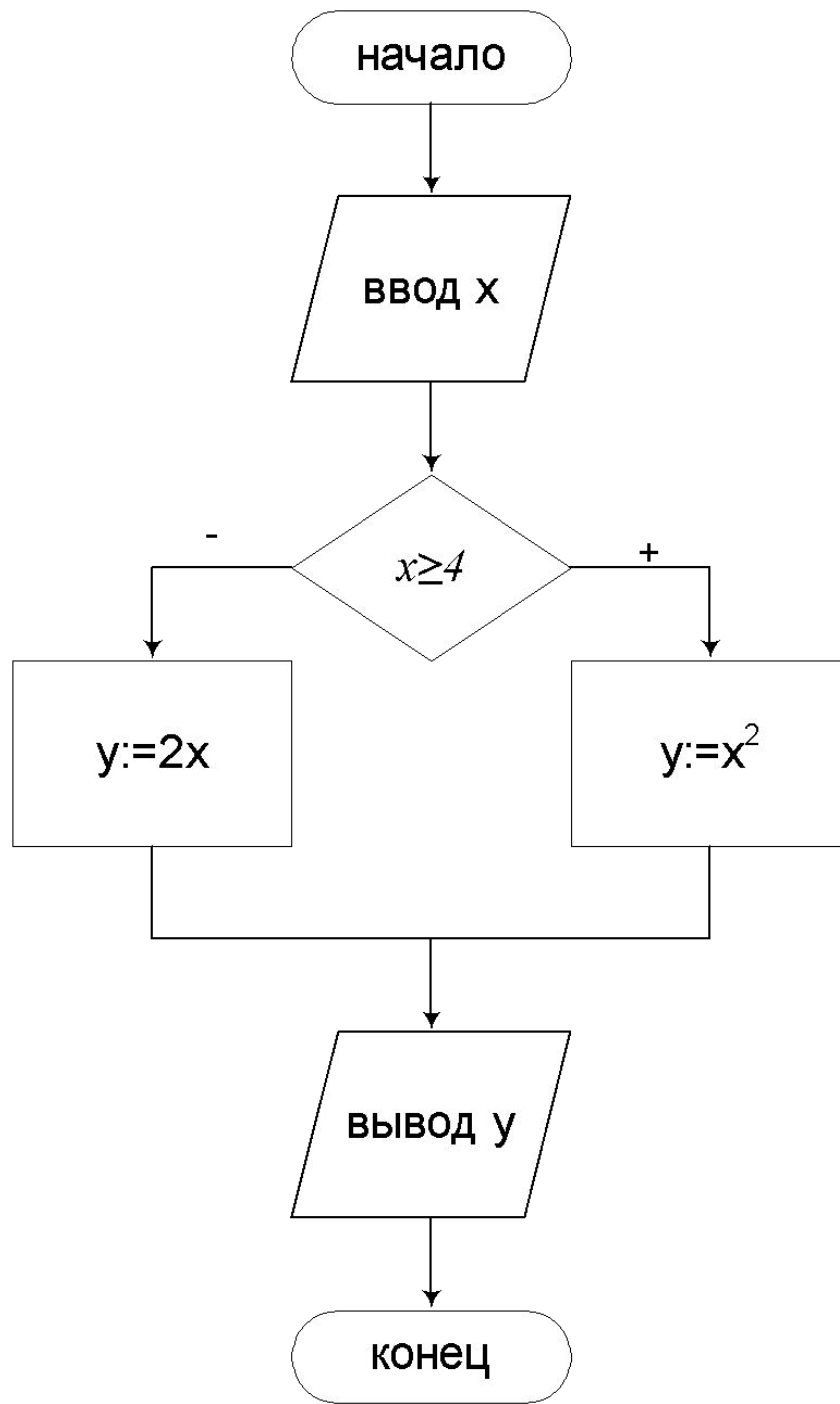
Найти: y

Решение:

Если $x \geq 4$, то $y = x^2$, иначе $y = 2x$

Пример 1

2. Блок-схема



Пример 1

Задание.

Составить программу вычисления значения функции

$$y = \begin{cases} x^2, & x \geq 4 \\ 2x, & x < 4 \end{cases}$$

3. Программа

```
program shoe;
```

4.

Результат

```
Окно вывода  
x=10  
y=100.0000
```

```
Окно вывода  
x=-3  
y=-6.0000
```

Пример 1

Задание.

Составить программу вычисления значения функции

$$y = \begin{cases} x^2, & x \geq 4 \\ 2x, & x < 4 \end{cases}$$

5. Таблица исполнения

Оператор	В памяти	На экране	Действия
<code>write('x=');</code>			
<code>read(x);</code>			
<code>if x>=4 then y:=sqr(x) else y:=2*x;</code>			
<code>write('y=',y:0:4)</code>			
<code>end.</code>			

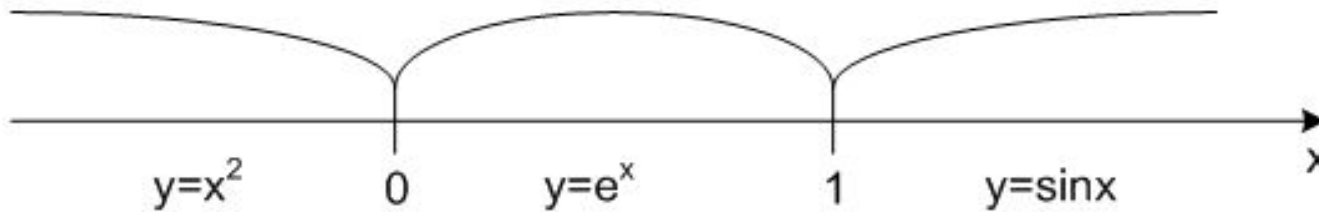
Пример 2

Задание.

Составить программу вычисления значения функции

$$y = \begin{cases} x^2, & \text{если } x \leq 0 \\ e^x, & \text{если } 0 < x < 1 \\ \sin x, & \text{если } x \geq 1 \end{cases}$$

1. Математическая модель



Дано: x

Найти: y

Решение:

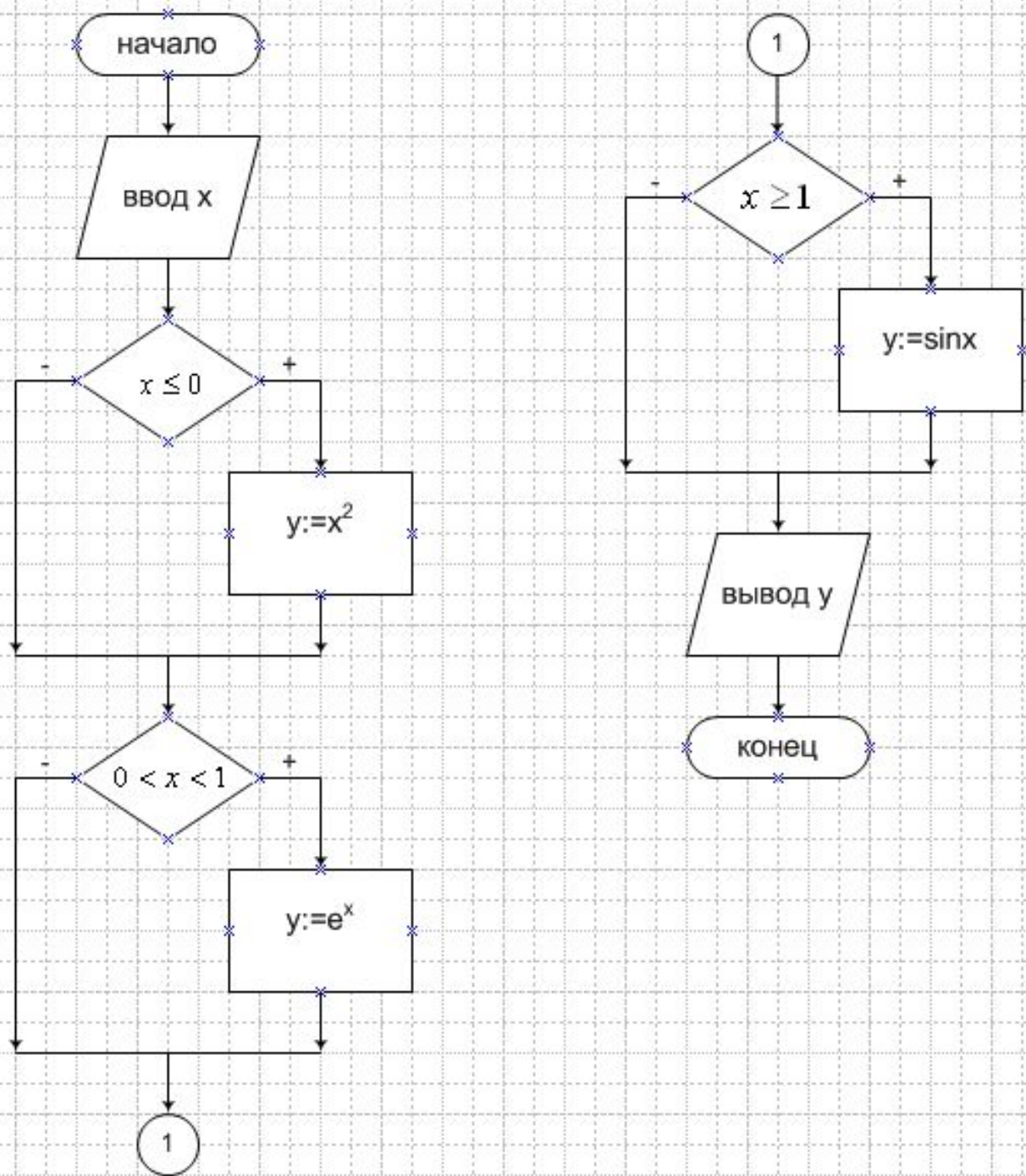
Если $x \leq 0$, то $y = x^2$

Если $x > 0$ и $x < 1$, то $y = e^x$

Если $x \geq 1$, то $y = \sin x$

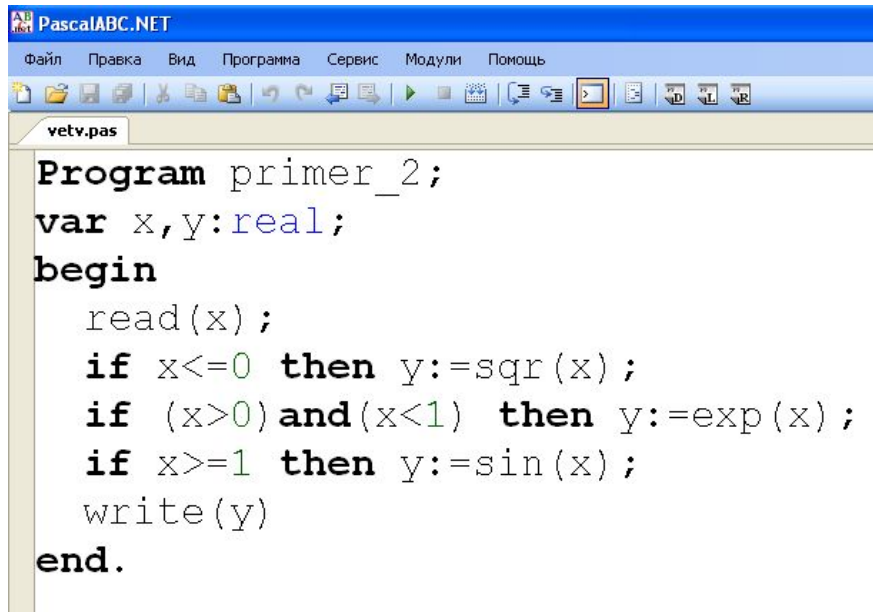
Пример 2

2. Блок-схема



Пример 2

3. Программа



```
Program primer_2;  
var x,y:real;  
begin  
  read(x);  
  if x<=0 then y:=sqr(x);  
  if (x>0) and(x<1) then y:=exp(x);  
  if x>=1 then y:=sin(x);  
  write(y)  
end.
```

4.

Результат



```
Окно вывода  
0.5  
1.64872127070013
```

Сложные ветвления

В каждой ветви оператора IF по умолчанию может выполняться только один оператор.

Если необходимо выполнить несколько операторов по условию, то их заключают в «операторные скобки» Begin...end.

```
If <условие> then
    begin
<оператор 1> ;
        <
оператор 2>;
        <
оператор 3>;
        end
    else
        begin
        <
оператор 4>;
        <
оператор 5>;
        <
оператор 6>;
```

Вложенные ветвления

Часто в программах приходится вычислять по нескольким функциям в зависимости от истинности или ложности проверяемых логических условий. В этих случаях в программах применяют несколько операторов IF – либо несколько «полных ветвлений», либо комбинацию из «коротких» и «полных ветвлений»

Для того, чтобы сократить количество проверяемых логических условий и правильнее записывать алгоритм программы, необходимо применять так называемые «вложенные» конструкции оператора IF

Вложенная форма ветвления:

```
If <условие1> then <оператор1>
    else
        if <условие2> then <оператор2>
            else
                if <условие3> then <оператор3>
                    else
                        if <условие4> then <оператор4>;
```

Замечание. Любой Else относится к последнему незакрытому IF