

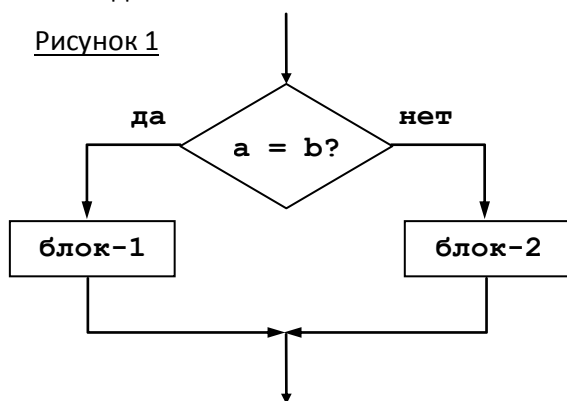
С1 (повышенный уровень, время – 30 мин)

Тема: Исправление ошибок в простой программе с условными операторами.

Что нужно знать:

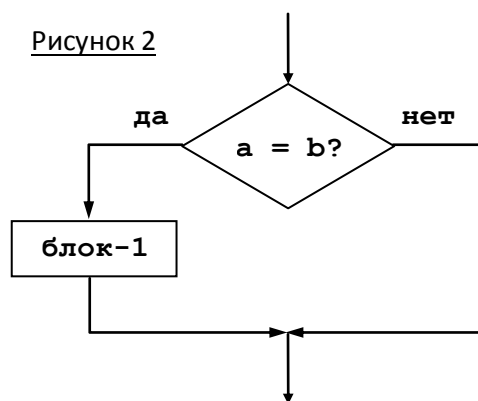
- правила построения программы на Паскале¹, Бэйсике или Си
- правила работы с переменными (объявление, ввод, вывод, оператор присваивания)
- *ветвление* – это выбор одного из двух возможных вариантов действий в зависимости от того, выполняется ли некоторое условие;
- на блок-схеме алгоритма ветвление изображается в виде блока-ромба с одним входом и двумя выходами:

Рисунок 1



полная форма ветвления

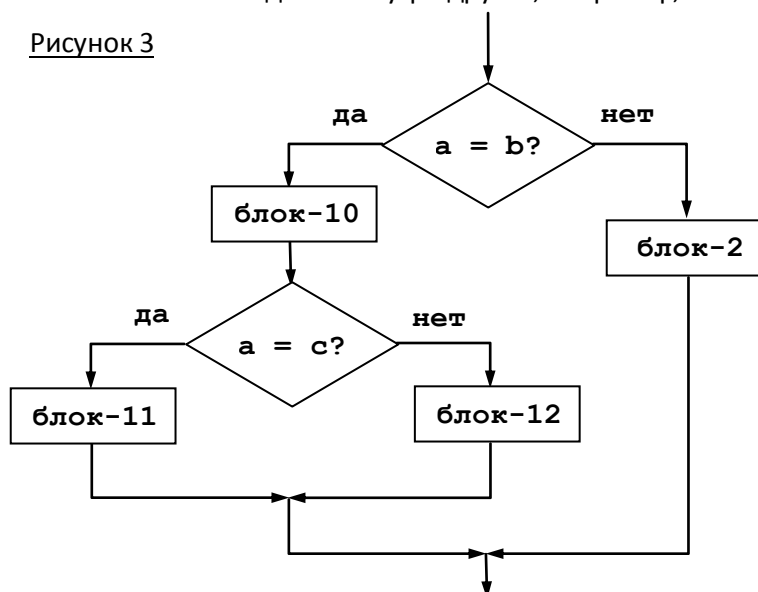
Рисунок 2



неполная форма ветвления

- если условие, записанное внутри ромба, истинно (ответ «да» на вопрос « $a=b$?»), выполняются все команды, входящие в **блок-1** (ветка «да»), иначе (если условие ложно) выполняются все команды в **блоке-2** (ветка «нет»)
- в неполной форме условного оператора **блок-2** пустой (отсутствует); теоретически можно сделать наоборот – так, чтобы **блок-1** оказался пустым, но это очень неграмотное решение, поскольку оно осложняет понимание алгоритма, запутывает его
- одна команда ветвления может находиться внутри другой, например, так:

Рисунок 3



- на этой схеме **блок-10** выполняется, когда $a=b$; **блок-11** – когда $a=b=c$, **блок-12** – когда $a=b$, но $a \neq c$ и, наконец, **блок-2** – когда $a \neq b$

¹ Далее при разборе задачи используется язык Паскаль, который наиболее распространен в школах России.

- на этой схеме (Рисунок 3) одна команда ветвления (с условием «**a=c**») вложена в другую (с условием «**a=b**»), каждая из них – это ветвление в полной форме; если блок-12 будет пустой (отсутствует), внутреннее ветвление имеет неполную форму; аналогично, если блок-2 пустой, то внешнее ветвление имеет неполную форму
- условный оператор **if-else** служит для организации ветвления в программе на языке Паскаль
- условный оператор может иметь полную или неполную форму; вот фрагменты программы, реализующие ветвления, показанные на рисунках 1 и 2:

полная форма:

```
if a = b then begin
    { блок-1 }
end
else begin
    { блок-2 }
end;
```

неполная форма:

```
if a = b then begin
    { блок-1 }
end;
```

здесь вместо комментариев в фигурных скобках (они выделены синим цветом) могут стоять любые операторы языка программирования (в том числе операторы присваивания, другие условные операторы, циклы, вызовы процедур и т.п.)

- обычно при записи программы операторы, находящиеся внутри обоих блоков, сдвигают вправо на 2-3 символа (запись «лесенкой»), это позволяет сразу видеть начало и конец блока (конечно, если «лесенка» сделана правильно)
- после **else** не надо (нельзя!) ставить какое-то условие, эта часть выполняется тогда, когда условие после **if** неверно (частая ошибка – после **else** пытаются написать условие, обратное тому, которое стоит после соответствующего ему **if**)
- в Паскале перед **else** не ставится точка с запятой, поскольку это ключевое слово обозначает не начало нового оператора, а вторую часть условного оператора **if-else**
- слова **begin** и **end** (их называют также «операторные скобки») ограничивают **блок-1** и **блок-2**; если внутри блока всего один оператор, эти «скобки» можно не писать, например, допустимы такие операторы

```
if a = b then
    c:=1
else c:=0;
```

```
if a = b then begin
    c:=1;
end
else c:=0;
```

```
if a = b then c:=1;
```

- а вот такие операторы недопустимы

```
if a = b then begin
    c:=1
else c:=0;
```

```
if a = b then
    c:=1;
end
else c:=0;
```

```
if a = b then
    c:=1;
    d:=1;
else x:=1;
```

- в первом случае есть **begin**, но забыли про соответствующий ему **end**;
- во втором фрагменте наоборот, есть **end**, а **begin** отсутствует;
- третий случай более сложный: судя по записи «лесенкой», здесь внутри блока-1 находятся 2 оператора, а операторных скобок **begin-end** нет; в результате получилось, что оператор **c:=1** находится внутри блока-1, он выполняется только при условии **a=b**; оператор **d:=1** выполняется всегда, после того, как условный оператор закончил работу; а **else** вообще «висит» непонятно как, тут транслятор выдаст ошибку; исправить эту программу можно так, как показано справа (добавив пару **begin-end**):

```
if a = b then begin
    c:=1;
    d:=1;
end
else x:=1;
```

- условный оператор может находиться внутри другого условного оператора, как в блоке-1, так и в блоке-2; например, схема на Рисунке 3 может быть записана на Паскале так:

```

if a = b then begin
  { блок-10 }
  if a = c then begin
    { блок-11 }
  end
  else begin
    { блок-12 }
  end;
end
else begin
  { блок-2 }
end;

```

- ключевая тема этого задания ЕГЭ – использование вложенных условных операторов, причем в тексте задания фрагмент программы обычно записан без отступов «лесенкой» или с неправильными отступами, например, так:

```

if a = b then begin
  if a = c then
    c:=1;
  end
  else c:=0;

```

```

if a = b then
  if a = c then
    c:=1
  else c:=0;

```

Чтобы разобраться с работой этих программ, нужно определить, к какому из условных операторов **if** относится часть **else**; для этого используют такое правило: «любой **else** относится к ближайшему **if**».

Рассмотрим фрагмент слева, в нем перед **else** стоит **end**, поэтому для него нужно найти соответствующий ему **begin**; таким образом определяем, что **else** относится к первому (внешнему) условному оператору.

В правом фрагменте перед **else** нет **end**, поэтому он относится к ближайшему по тексту внутреннему условному оператору.

Блок-схемы для двух фрагментов показаны ниже, желтым цветом выделен «переехавший» блок:

Рисунок 4

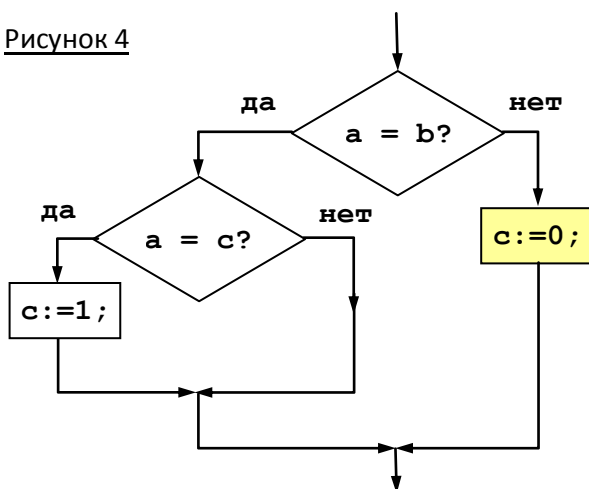
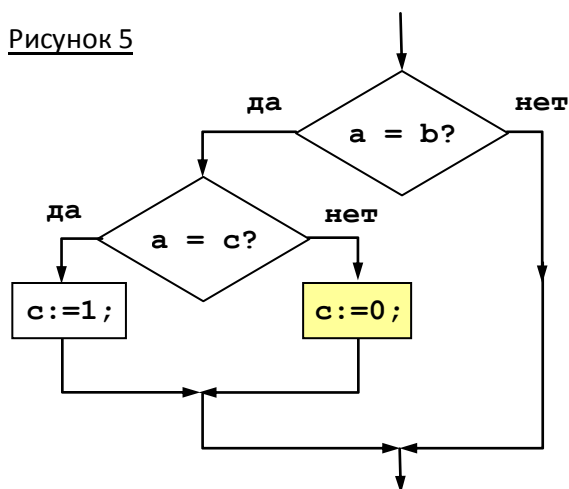


Рисунок 5



- в условных операторах можно использовать сложные условия, которые строятся из простых отношений (<, <=, >, >=, =, <>) с помощью логических операций **not** («НЕ», отрицание), **and** («И», одновременное выполнение двух условий) и **or** («ИЛИ», выполнение хотя бы одного из двух условий)

- в сложном условии сначала выполняются действия в скобках, потом – **not**, затем – **and**, затем – **or** и, наконец, отношения;
операции равного уровня (приоритета) выполняются последовательно слева направо
- поскольку отношения в Паскале имеют низший приоритет, в сложном условии их приходится брать в скобки:

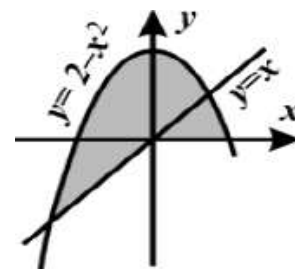
```
if (a = b) or (b < c) and (c <> d) then begin
    ...
end;
```

- в приведенном выше примере сначала определяются результаты сравнения (выражения в скобках), затем выполняется операция **and** («И»), а затем – **or** («ИЛИ»)

Пример задания:

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x , y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x,y: real;
begin
  readln(x,y);
  if y>=x then
    if y>=0 then
      if y<=2-x*x then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
end.
```

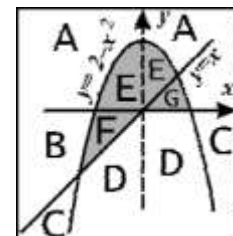


Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F и G). Точки, лежащие на границах областей, отдельно не рассматривать.

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).



Решение:

- 1) начнем заполнять таблицу, выписывая истинность каждого из трёх условий
- 2) условие $y \geq x$ истинно выше прямой $y=x$, то есть в областях A, B, E, F

Область	$y \geq x$?	$y \geq 0$?	$y \leq 2 - x^2$?	вывод	верно?
A	да				
B	да				

C	нет				
D	нет				
E	да				
F	да				
G	нет				

- 3) условие $y \geq 0$ истинно выше прямой $y=0$, то есть в областях A, E, G, однако **это условие проверяется только тогда, когда первое условие, $y \geq x$, истинно**; поэтому для всех областей, где первое условие неверно (это области C, D, G), сразу в столбце второго условия ставим прочерк (условие не будет проверяться)

Область	$y \geq x$?	$y \geq 0$?	$y \leq 2 - x * x$?	вывод	верно?
A	да	да			
B	да	нет			
C	нет	–			
D	нет	–			
E	да	да			
F	да	нет			
G	нет	–			

- 4) третье условие выполняется для областей «внутри» параболы, то есть для E, F, G, D; однако оно проверяется только тогда, когда первые два истинны (для A и E), в остальных строках ставим прочерк:

Область	$y \geq x$?	$y \geq 0$?	$y \leq 2 - x * x$?	вывод	верно?
A	да	да	нет		
B	да	нет	–		
C	нет	–	–		
D	нет	–	–		
E	да	да	да		
F	да	нет	–		
G	нет	–	–		

- 5) как следует из текста программы, она выведет что-то на экран только в том случае, когда выполняются первые два условия и программа выходит на третье: для области A будет выведено «не принадлежит», для области E – «принадлежит», именно в этих двух случаях программа работает правильно, в остальных – нет:

Область	$y \geq x$?	$y \geq 0$?	$y \leq 2 - x * x$?	вывод	верно?
A	да	да	нет	не принадлежит	да
B	да	нет	–	–	нет
C	нет	–	–	–	нет
D	нет	–	–	–	нет
E	да	да	да	принадлежит	да
F	да	нет	–	–	нет
G	нет	–	–	–	нет

- 6) для того, чтобы доработать программу, проще всего составить одно сложное условие, описывающее всю заштрихованную область
- 7) в данном случае удобно представить данную область в виде объединения областей, первая из которых включает области E+G, а вторая – области E+F
- 8) область E+G соответствует условию $(y \geq 0)$ and $(y \leq 2 - x * x)$
- 9) область E+F соответствует условию $(y \geq x)$ and $(y \leq 2 - x * x)$
- 10) объединение областей выполняется с помощью операции ИЛИ (or), так что полное условие принимает вид

$$(y \geq 0) \text{ and } (y \leq 2 - x * x) \text{ or } (y \geq x) \text{ and } (y \leq 2 - x * x)$$

поскольку операция И (**and**) имеет более высокий приоритет, чем ИЛИ (**or**), порядок выполнения операций тут правильный; в случае сомнений можно поставить дополнительные скобки:

$((y \geq 0) \text{ and } (y \leq 2 - x * x)) \text{ or } ((y \geq x) \text{ and } (y \leq 2 - x * x))$

11) поскольку в обоих условиях есть условие $y \leq 2 - x * x$, запись можно немного сократить:

$(y \leq 2 - x * x) \text{ and } ((y \geq x) \text{ or } (y \geq 0))$

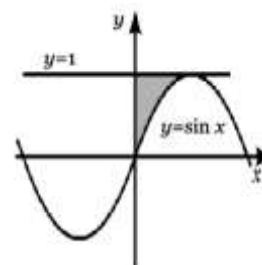
12) доработанная программа выглядит так:

```
var x,y: real;
begin
  readln(x,y);
  if (y <= 2-x*x) and ((y >= x) or (y >= 0)) then
    write('принадлежит')
  else
    write('не принадлежит')
  end.
```

Ещё пример задания:

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x,y: real;
begin
  readln(x,y);
  if y <= 1 then
    if x >= 0 then
      if y >= sin(x) then
        write('принадлежит')
      else write('не принадлежит')
    end.
```



Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

Решение:

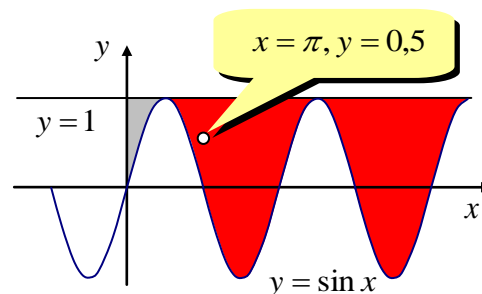
- 1) сначала лучше отложить в сторону программу и попытаться написать условие, которым должны отвечать точки, попавшие в выделенную область
- 2) заштрихованная область ограничена по координате x , она находится
 - справа от оси y , что равносильно **условию** $x \geq 0$ (с учетом границы здесь и далее получаем нестрогие неравенства)
 - слева от первого максимума функции $y = \sin x$; из математики мы знаем, что эта функция достигает максимума при $x = \frac{\pi}{2}$, поэтому получаем **второе условие** $x \leq \frac{\pi}{2}$
- 3) заштрихованная область ограничена с двух сторон по координате y : она находится
 - ниже линии $y = 1$, откуда следует **третье условие** $y \leq 1$
 - выше линии $y = \sin x$, что дает **четвертое условие** $y \geq \sin x$

- 4) итак, точка находится в заданной области, если все эти четыре условия выполняются одновременно; можно предположить, что в программе нужно использовать четыре вложенных условных оператора или один условный оператор, в котором четыре простых условия (отношения $x \geq 0$, $x \leq \frac{\pi}{2}$, $y \leq 1$ и $y \geq \sin x$) связаны с помощью логической операции **and** («И», одновременное выполнение всех условий)
- 5) теперь смотрим на программу: здесь три (а не четыре!) вложенных условных оператора с простыми отношениями, поэтому явно какое-то условие не учтено; легко найти, что «забыли» условие $x \leq \frac{\pi}{2}$

- 6) оператор **write('принадлежит')** помещен внутрь всех трех условных операторов, то есть, он выполнится тогда, когда три (а не четыре!) условия истинны;

- 7) отметим на рисунке область, где выполняются все нужные условия, кроме $x \leq \frac{\pi}{2}$ (красная зона);

- 8) для всех точек, которые находятся в «красной» зоне программа выдаст сообщение «принадлежит», хотя в самом деле эти точки не принадлежат заданной области; одна из таких точек имеет координаты $x = \pi, y = 0,5$



- 9) теперь выясним, когда программа выдает сообщение «не принадлежит»

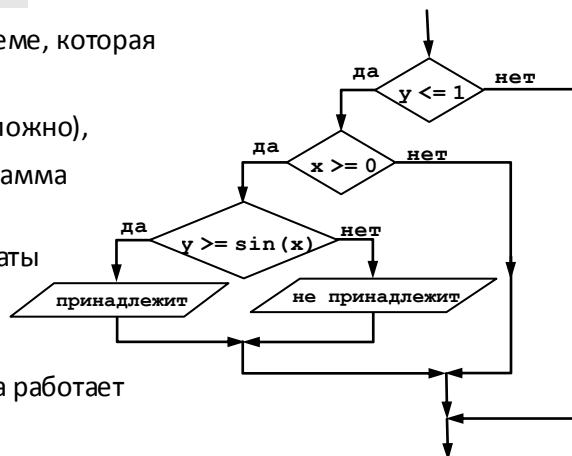
```
if y <= 1 then
  if x >= 0 then
    if y >= sin(x) then
      write('принадлежит')
    else write('не принадлежит')
```

- 10) судя по записи «лесенкой», **else** относится к самому первому оператору **if**, однако в самом деле это не так; перед словом **else** нет **end**, поэтому ищем ближайший **if**: это самый внутренний оператор, правильная запись «лесенкой» выглядит так:

```
if y <= 1 then
  if x >= 0 then
    if y >= sin(x) then
      write('принадлежит')
    else write('не принадлежит')
```

- 11) этот фрагмент программы соответствует блок-схеме, которая показана на рисунке справа:

- 12) по схеме видим, что при $y > 1$ (первое условие ложно), а также при $x < 0$ (второе условие ложно) программа вообще не выдает никакого сообщения, то есть, работает неправильно; таким образом, координаты любой точки, для которой $y > 1$ или $x < 0$, могут быть указаны в ответе как пример набора входных данных, при которых программа работает неправильно



- 13) итак, первая часть ответа такова

примеры входных данных, на которых программа работает неверно:

($x=3.14$, $y=0.5$) (неправильно определяет принадлежность точки области)

($x=0$, $y=1$) или ($x=-1$, $y=0$) (не выдает вообще никакого сообщения)

14) остается исправить эту программу;

начнем с самого «лобового способа»: добавим в программу четвертый (вложенный)

условный оператор, проверяющий условие $x \leq \frac{\pi}{2}$, и еще три блока **else**, чтобы выводить

строку «не принадлежит» в том случае, когда хотя бы один из них не сработал:

```
if x <= pi/2 then
  if y <= 1 then
    if x >= 0 then
      if y >= sin(x) then
        write('принадлежит')
      else write('не принадлежит')
    else write('не принадлежит')
  else write('не принадлежит')
else write('не принадлежит');
```

обратите внимание, что точка с запятой есть только после самого последнего оператора **write**, так как остальные стоят перед ключевым словом **else**, перед которым точка с запятой не ставится

15) хотя приведенный выше метод дает работоспособную программу, она получается слишком длинная и некрасивая для такой простой задачи; достаточно сказать, что оператор

write('не принадлежит') повторяется в тексте 4 раза

16) более элегантное решение формулируется на словах так: «точка принадлежит области, если выполняются одновременно 4 приведенных выше условия, а иначе – не принадлежит»; а вот реализация на Паскале (приведем программу-ответ целиком):

```
var x,y: real;
begin
  readln(x,y);
  if (x >= 0) and (x <= pi/2) and
    (y <= 1) and (y >= sin(x)) then
    write('принадлежит')
  else write('не принадлежит');
end.
```

здесь использовано сложное условие, в котором 4 отношения связаны операциями **and** («И», требуется одновременное выполнение всех условий)

Возможные проблемы:

- как правило, в этой задаче требуется использовать знания из курса математики (решение уравнений, графики функций, область допустимых значений, составление уравнений прямой по приведенному графику)
- как показывает анализ опубликованных задач этого типа, нужно уметь, прежде всего, разбираться в серии вложенных условных операторов в полной и неполной форме
- неправильная «лесенка» в записи сбивает с толку и подталкивает к неверному решению; чтобы разобраться в программе, лучше на черновике построить блок-схему алгоритма и правильную «лесенку»
- чтобы не запутаться, к какому оператору относится **else**, используйте следующее правило:

- если перед **else** нет слова **end**, нужно искать ближайший сверху условный оператор **if**
- если перед **else** стоит **end** (конец блока), нужно искать парный ему **begin** (начало блока) и соответствующий условный оператор **if ... then begin**
- проверяйте, все ли необходимые условия учтены в программе, это особенно актуально для *немонотонных* функций типа синуса или косинуса (немонотонные функции на некоторых участках возрастают при увеличении аргумента, а на некоторых – убывают);
например, в этой задаче можно пропустить необходимость выполнения условия $x \leq \frac{\pi}{2}$
- не перепутайте, где нужно использовать операцию **and** («И», одновременное выполнение условий), а где – **or** («ИЛИ», хотя бы одно условие)
- нужно внимательно проверять, всегда ли программа выдает сообщение, если заданное условие не выполняется
- часто бывает полезно нарисовать блок-схему алгоритма, которая позволяет увидеть ход выполнения программы при всех возможных вариантах
- проверяйте, включает ли заданная область свои границы; если включает – в отношениях будут нестрогие неравенства (\leq , \geq), если не включает – строгие ($<$, $>$)
- при оценке работы можно (при абсолютно правильном решении) потерять баллы из-за синтаксических ошибок в программе (скобки, точки с запятой, неправильное написание оператора и т.п.); не забывайте, что
 - в сложном условии все простые условия (отношения) нужно брать в скобки, так как в Паскале отношения при вычислении логического выражения имеют самый низкий приоритет
 - перед **else** точка с запятой никогда не ставится
 - в конце программы после последнего **end** ставится точка

За что снимают баллы:

- неправильно определены входные данные, при которых исходная программа работает неверно
- исправлены не все ошибки в программе, например, легко «просмотреть», что необходимо еще условие $x \leq \frac{\pi}{2}$
- программа работает правильно в большем количестве случаев, чем исходная, но не для всех возможных исходных данных
- перепутаны знаки $<$ и $>$, логические операции **or** и **and**
- неверно расставлены операторные скобки **begin-end**
- синтаксические ошибки (знаки пунктуации – запятые, точки, точки с запятой; неверное написание ключевых слов); чтобы получить 3 балла, нужно при абсолютно правильном решении сделать не более одной синтаксической ошибки; на 2 балла – до двух ошибок, на 1 балл – до трех ошибок

Задачи для тренировки²:

- 1) Требовалось написать программу, которая решает уравнение « $a|x| = b$ » относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно:

```
var a,b,x: real;
begin
  readln(a,b,x);
  if a = 0 then
    if b = 0 then
      write ('любое число')
    else write ('нет решений')
    else
      if b = 0 then
        write('x = 0')
      else write('x = ',b/a, ' или x = ', -b/a);
    end.
```

Последовательно выполните три задания: 1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу. 2) Укажите, какая часть программы является лишней. 3) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 2) Требовалось написать программу, которая решает уравнение « $ax + b = 0$ » относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно:

```
var a, b, x: real;
begin
  readln(a,b,x);
  if b = 0 then
    write('x = 0')
  else
    if a = 0 then
      write('нет решений')
    else
      write('x = ', -b/a);
    end.
```

Последовательно выполните три задания: 1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу. 2) Укажите, какая часть программы является лишней. 3) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

² Источники заданий:

1. Демонстрационные варианты ЕГЭ 2004-2011 гг.
2. Гусева И.Ю. ЕГЭ. Информатика: раздаточный материал тренировочных тестов. — СПб: Тригон, 2009.
3. Крылов С.С., Лещинер В.Р., Якушкин П.А. ЕГЭ-2010. Информатика. Универсальные материалы для подготовки учащихся / под ред. В.Р. Лещинера / ФИПИ. — М.: Интеллект-центр, 2010.
4. Абрамян М.Э., Михалкович С.С., Русанова Я.М., Чердынцева М.И. Информатика. ЕГЭ шаг за шагом. — М.: НИИ школьных технологий, 2010.
5. Тренировочные и диагностические работы МИОО 2010-2012.

- 3) Требовалось написать программу, которая определяет, лежит ли точка $A(x_0, y_0)$ внутри треугольной области, ограниченной осями координат и прямой $y = 2 - x$ («внутри» понимается в строгом смысле, т.е. случай, когда точка A лежит на границе области, недопустим). В результате программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

```
var x0, y0, y: real;
begin
  readln (x0, y0);
  if (x0 < 2) then begin
    if (x0 > 0) then begin
      y = 2 - x0;
      if (y0 < y) then
        writeln ('точка лежит внутри области')
      else writein ('точка не лежит внутри области');
    end
  else writeln ('точка не лежит внутри области');
  end
  else writeln ('точка не лежит внутри области');
  end.
```

Последовательно выполните задания:

1. Приведите пример таких чисел x_0 и y_0 , при которых программа неверно решает поставленную задачу.
 2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).
 3. Укажите, как можно доработать программу, чтобы вместо вложенных операторов **IF** она содержала логическую операцию **AND**.
- 4) Требовалось написать программу, которая решает уравнение $ax^2 + bx + c = 0$ относительно x для действительных чисел a, b, c , введенных с клавиатуры, о которых заведомо известно, что $a \neq 0$, $b \neq 0$ и $c \neq 0$. Была написана следующая программа:

```
var a, b, c, D, x1, x2: real;
begin
  readln(a, b, c, x1, x2);
  D := b*b - 4*a*c;
  if D > 0
  then begin
    x1 := (-b + sqrt(D)) / (2*a);
    x2 := (-b - sqrt(D)) / (2*a);
    write('x1 =', x1);
    write('x2 =', x2); end
  else writeln ('действительных корней нет');
  end.
```

Известно, что программа написана с ошибками. Последовательно выполните три задания:

1. Приведите пример таких чисел a, b, c , при которых программа неверно решает поставленную задачу.
2. Укажите, какая часть программы является лишней.

3. Укажите, как, по-вашему мнению, нужно доработать программу, чтобы не было случаев ее неправильной работы.

- 5) Требовалось написать программу, которая определяет, имеется ли среди введенных с клавиатуры положительных целых чисел **a** и **b** хотя бы одно четное. Была написана следующая программа:

```
var a, b: integer;
begin
  readln(a, b);
  a := a mod 2;
  if a > 0 then b := b mod 2;
  if b > 0 then
    writeln ('четных чисел нет')
  else writeln ('четное число есть');
end.
```

Известно, что программа написана с ошибками. Последовательно выполните три задания:

1. приведите пример таких чисел **a**, **b**, при которых программа неверно решает поставленную задачу;
2. укажите, как, по вашему мнению, нужно доработать программу, чтобы не было случаев ее неправильной работы;
3. укажите, как можно доработать программу, чтобы она вместо вложенных операторов **IF** содержала логическую операцию **OR**.

- 6) Требовалось написать программу, которая определяет, можно ли построить треугольник из отрезков с длинами **x**, **y**, **z**. Программа должна выводить соответствующее текстовое сообщение.

Программист сделал в программе ошибки.

```
var x, y, z: real;
begin
  readln (x, y, z);
  if (x + y > z) then
    begin
      if (x + z > y) then
        if (y + z > x) then
          writeln('треугольник построить можно');
        end
      else writeln('треугольник построить нельзя');
    end
  end
```

Последовательно выполните задания:

1. Приведите пример таких чисел **x**, **y**, **z**, при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать один из способов доработки исходной программы).
3. Укажите, как можно доработать программу, чтобы она вместо вложенных операторов **IF** содержала логическую операцию **AND**.

- 7) Требовалось написать программу, которая определяет, лежит ли точка **A(x, y)** внутри некоторого кольца («внутри» понимается в строгом смысле, т.е. случай, когда точка **A** лежит на границе кольца, недопустим). Центр кольца находится в начале координат. Для кольца заданы внутренний и внешний радиусы **r1**, **r2**; известно, что **r1** отлично от **r2**, но неизвестно, **r1 > r2** или **r2 > r1**. В том случае, когда точка **A** лежит внутри кольца, программа должна выводить

соответствующее сообщение, в противном случае никакой выходной информации не выдается.

Программист сделал в программе ошибки.

```
var x, y, r1, r2: real;
    h: real;
begin
  readln (x, y, r1, r2);
  h := sqrt (x*x + y*y);
  if (h < r1) AND (h > r2) then
    writeln ('точка A лежит внутри кольца');
  end.
```

Последовательно выполните задания:

1. Приведите пример таких чисел x и y , при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).
3. Укажите, как можно доработать программу, чтобы она не содержала логических операций **AND** или **OR**.

Примечание: для обозначения расстояния от точки **A** до начала координат используется вспомогательная переменная **h**.

- 8) Требовалось написать программу, которая определяет, лежит ли точка **A** (x_0, y_0) внутри квадратной области, ограниченной прямыми, параллельными осям координат: $y = -2, y = 2, x = -2, x = 2$ («внутри» понимается в строгом смысле, т.е. случай, когда точка **A** лежит на границе области, недопустим).

В результате программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

```
var x0, y0: real;
begin
  readln (x0, y0)
  if (x0 < 2) then
    begin
      if (x0 > -2) then
        if (y0 < 2) then
          writeln('точка лежит внутри области');
        end
      else writeln('точка не лежит внутри области');
    end
  end.
```

Последовательно выполните задания:

1. Приведите пример таких чисел x_0 и y_0 , при которых программа неверно решает поставленную задачу.
 2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).
 3. Укажите, как можно доработать программу, чтобы она вместо вложенных операторов **IF** содержала логическую операцию **AND**.
- 9) Рассматривается стандартная шахматная доска размером 8x8. Примем, что i – номер вертикали (может принимать значения от 1 до 8), j – номер горизонтали (также может принимать значения от 1 до 8). В левом нижнем углу, т.е. на поле $i=1, j=1$ (это поле черного цвета) стоит черный король. Напомним, что король может ходить на 1 клетку в любом направлении (по горизонтали, вертикали или диагонали). В правом нижнем углу, т.е. на поле $i=8, j=1$ (это поле белого цвета)

стоит белый король. Введены обозначения: $P(i, j)$ минимальное число ходов, за которое черный король может попасть на поле (i, j) ; $V(i, j)$ минимальное число ходов, за которое белый король может попасть на поле (i, j) .

Программист написал программу, в которой требовалось определить все такие поля (i, j) , для которых $P(i, j) = V(i, j)$, и выдать на экран соответствующие значения i, j (текст программы приведен ниже).

```
var i, j: integer;
begin
  writeln('искомые поля');
  for j:=5 to 8 do
    for i:=1 to 8 do
      begin
        if (i=9-j) OR (i=j)
          then writeln('i=', i, ' j=', j);
      end;
    end.
end.
```

Последовательно ответьте на вопросы:

1. Выдаст ли программа, написанная программистом, поле, для которого $i=4, j=5$?
2. Указать все из перечисленных ниже полей, которые удовлетворяют постановке задачи, т.е. для таких полей должно быть выполнено $P(i, j) = V(i, j)$:
 $(i=1, j=8), (i=2, j=8), (i=1, j=7), (i=5, j=5), (i=8, j=6)$
3. Видно, что программист допустил ошибку в программе. Укажите, какую доработку программы нужно провести, чтобы она соответствовала постановке задачи (такая доработка может быть проведена неединственным образом – годится любой правильный вариант доработки).

- 10) Требовалось написать программу, которая решает неравенство $\frac{x-a}{bx} > 0$ относительно x для

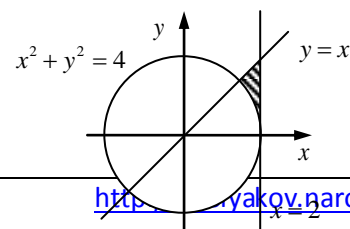
всех ненулевых действительных чисел $a \neq 0$ и $b \neq 0$, введенных с клавиатуры. Программист торопился и сделал в программе ошибки.

```
var a, b, x: real;
begin
  readln(a, b, x);
  if b > 0 then
    write('x > ', a, ' или x < 0')
  else
    if a > 0 then
      write('0 < x < ', a)
    else
      write(a, ' < x < 0');
  end.
```

Последовательно выполните три задания:

1. Приведите примеры таких чисел a, b, x , при которых программа неверно решает поставленную задачу.
2. Укажите, какая часть программы является лишней?
3. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

- 11) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области,



включая ее границы. Программист торопился и написал программу неправильно. Вот она:

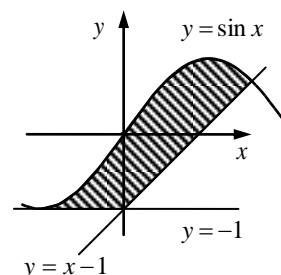
```
var x,y: real;
begin
  readln(x,y);
  if x*x+y*y >= 4 then
    if x <= 2 then
      if y <= x then
        write('принадлежит')
      else
        write('не принадлежит')
      end.
    end.
```

Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 12) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно.

Вот она:

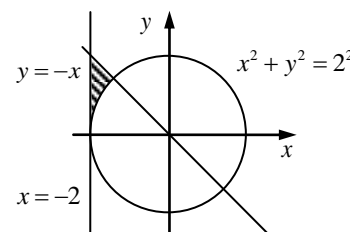
```
var x,y: real;
begin
  readln(x,y);
  if y >= -1 then
    if y <= sin(x) then
      if y >= x-1 then
        write('принадлежит')
      else
        write('не принадлежит')
      end.
    end.
```



Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 13) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if x*x + y*y >= 4 then
    if x >= -2 then
      if y <= -x then
        write('принадлежит')
      else
        write('не принадлежит')
      end.
```



```
write('не принадлежит')
end.
```

Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

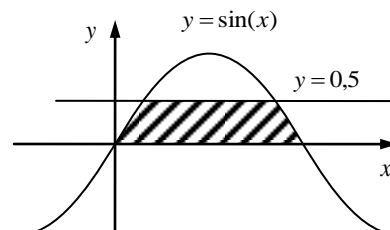
- 14) Требовалось написать программу, которая вводит три целых числа в переменные a, b и c , а потом сортирует их так, чтобы в переменной a оказалось наименьшее число, в переменной b – среднее число, а в переменной c – наибольшее число. Программист торопился и сделал в программе ошибки.

```
var a, b, c, x: integer;
begin
  read(a, b, c, x);
  if a > b then begin x:=a; a:=b; b:=x; end;
  if b > c then begin x:=b; b:=c; c:=x; end;
  write(a,b,c);
end.
```

Последовательно выполните три задания:

1. Приведите примеры таких чисел a, b, c , при которых программа неверно решает поставленную задачу.
 2. Укажите, какая часть программы является лишней?
 3. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).
- 15) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if y <= sin(x) then
    if y <= 0.5 then
      if y >= 0 then
        write('принадлежит')
      else
        write('не принадлежит');
      end;
    end;
  end.
```



Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 16) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

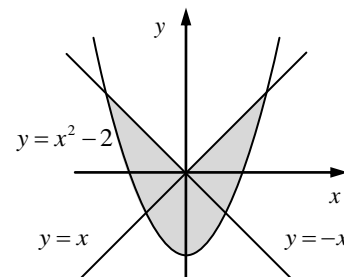
```
var x, y: real;
begin
```



```

readln(x, y);
if y <= x then
if y <= -x then
if y >= x*x-2 then
write('принадлежит')
else
write('не принадлежит');
end.

```



Последовательно выполните следующее: 1) Приведите пример

таких чисел x, y , при которых программа неверно решает

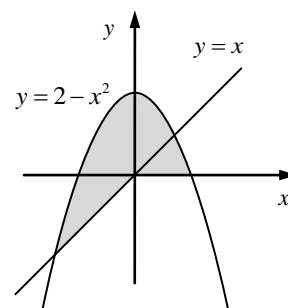
поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 17) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
readln(x, y);
if y >= x then
if y >= 0 then
if y <= 2-x*x then
write('принадлежит')
else
write('не принадлежит');
end.

```



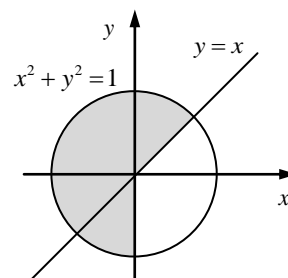
Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 18) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
readln(x, y);
if y >= x then
if x <= 0 then
if x*x+y*y <= 1 then
write('принадлежит')
else
write('не принадлежит');
end.

```



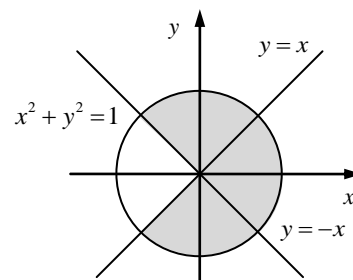
Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 19) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x , y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
  readln(x, y);
  if y <= x then
    if y >= -x then
      if x*x+y*y <= 1 then
        write('принадлежит')
      else
        write('не принадлежит');
      end.

```



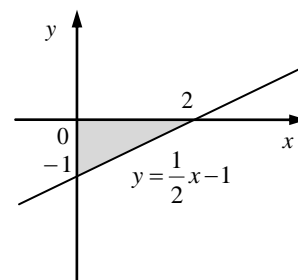
Последовательно выполните следующее: 1) Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 20) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x_0 , y_0 – действительные числа) и определяет принадлежность точки заштрихованной области, НЕ включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x0, y0, y: real;
begin
  readln(x0, y0);
  if x0 < 2 then
    if x0 > 0 then begin
      y := x0/2 - 1;
      if y0 < y then
        write('принадлежит')
      else
        write('не принадлежит')
      end
    end
  else
    write('не принадлежит');
  end.

```



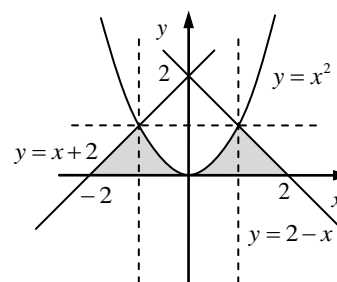
Последовательно выполните следующее: 1) Приведите пример таких чисел x_0 , y_0 , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 21) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x , y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
  readln(x, y);

```



```

if y <= x*x then
if x-y >= -2 then
if x+y <= 2 then
write('принадлежит')
else
write('не принадлежит');
end.

```

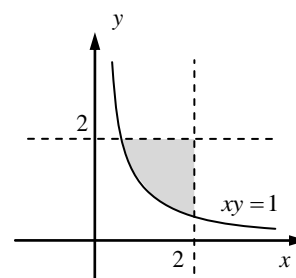
Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 22) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
readln(x, y);
if x*y >= 1 then
if x <= 2 then
if y <= 2 then
write('принадлежит')
else
write('не принадлежит');
end.

```



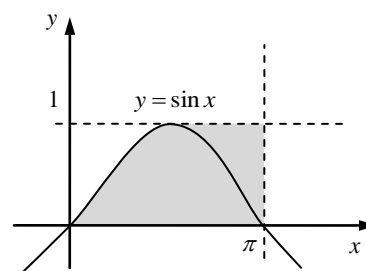
Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 23) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
readln(x, y);
if y >= 0 then
if y <= 1 then
if x >= 0 then
if x <= pi then
if y <= sin(x) then
write('принадлежит')
else
write('не принадлежит');
end.

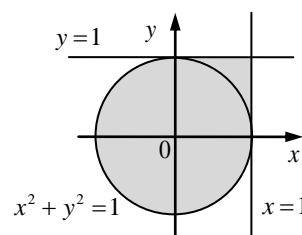
```



Последовательно выполните следующее: 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 24) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if x <= 1 then
    if y <= 1 then
      if x*x + y*y <= 1 then
        write('принадлежит')
      else
        write('не принадлежит');
      end.
    end.
```

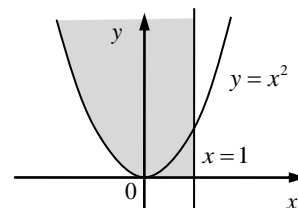


Последовательно выполните следующее:

- 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. **Объясните, почему для указанных чисел программа неверно решает поставленную задачу.**
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 25) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if y >= 0 then
    if x <= 1 then
      if y >= x*x then
        write('принадлежит')
      else
        write('не принадлежит');
      end.
    end.
```



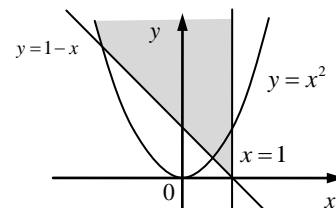
Последовательно выполните следующее:

- 1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. **Объясните, почему для указанных чисел программа неверно решает поставленную задачу.**

2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 26) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if x <= 1 then
    if y >= 1-x then
      if y >= x*x then
        write('принадлежит')
      else
        write('не принадлежит');
      end.
    end.
```



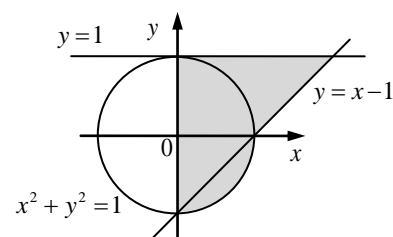
Последовательно выполните следующее:

1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. **Объясните, почему для указанных чисел программа неверно решает поставленную задачу.**

2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 27) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if x >= 0 then
    if y <= 1 then
      if (x*x + y*y <= 1) and (y >= x-1) then
        write('принадлежит')
      else
        write('не принадлежит');
      end.
    end.
```

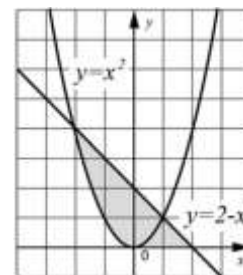


Последовательно выполните следующее:

1) Приведите пример таких чисел x, y , при которых программа неверно решает поставленную задачу. **Объясните, почему для указанных чисел программа неверно решает поставленную задачу.**

2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 28) Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу



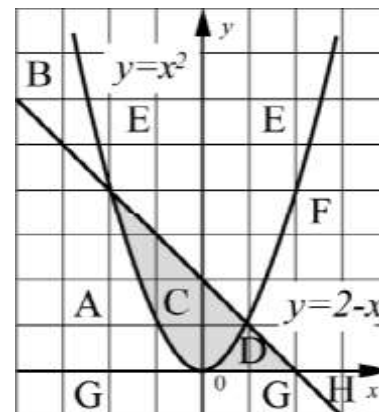
```

var x,y: real;
begin
  readln(x,y);
  if y<=2-x then
    if y>=0 then
      if y>=x*x then
        write('принадлежит')
      else
        write('не принадлежит')
    end.

```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.

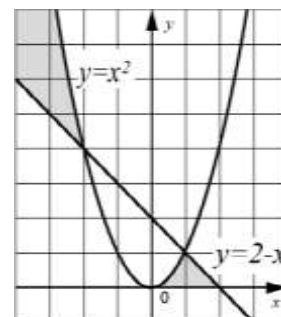


Область	$y \leq 2 - x$?	$y \geq 0$?	$y \geq x * x$?	вывод	верно?
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 29) Треб овалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу



```

var x,y: real;
begin
  readln(x,y);

```

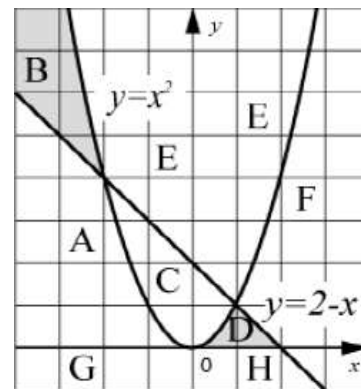
```

if y>=0 then
  if y>=2-x then
    if y<=x*x then
      write('принадлежит')
    else
      write('не принадлежит')
  end.

```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.



Область	$y \geq 0$?	$y \geq 2-x$?	$y \leq x*x$?	вывод	верно?
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

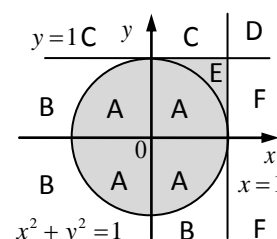
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 30) (С.Э. Назаренко) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```

var x, y: real;
begin
  readln(x, y);
  if x <= 1 then
    if y <= 1 then
      if x*x + y*y <= 1 then
        write('принадлежит')
      else
        write('не принадлежит');
    end.

```



1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F). Точки, лежащие на границах областей, отдельно не рассматривать.

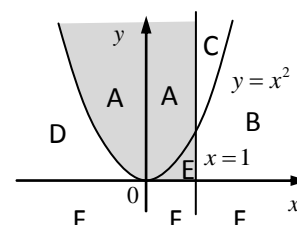
Область	$x \leq 1$	$y \leq 1$	$x*x + y*y \leq 1$	вывод	верно?
A					
B					
C					
D					
E					
F					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 31) (С.Э. Назаренко) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if y >= 0 then
    if x <= 1 then
      if y >= x*x then
        write('принадлежит')
      else
        write('не принадлежит');
      end.
```



1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F). Точки, лежащие на границах областей, отдельно не рассматривать.

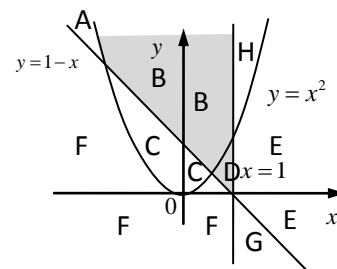
Область	$y \geq 0$	$x \leq 1$	$y \geq x*x$	вывод	верно?
A					
B					
C					
D					
E					
F					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 32) (С.Э. Назаренко) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if x <= 1 then
    if y >= 1-x then
      if y >= x*x then
        write('принадлежит')
      else
        write('не принадлежит');
    end.
```



1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G, H). Точки, лежащие на границах областей, отдельно не рассматривать.

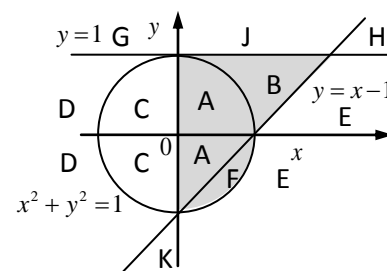
Область	$x \leq 1$	$y \geq 1-x$	$y \geq x^2$	вывод	верно?
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 33) (С.Э. Назаренко) Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
var x, y: real;
begin
  readln(x, y);
  if x >= 0 then
```



```

if y <= 1 then
  if (x*x + y*y <= 1) and (y >= x-1) then
    write('принадлежит')
  else
    write('не принадлежит');
  end.

```

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D, E, F, G, H, J, K). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	$x \geq 0$	$y \leq 1$	$x^2 + y^2 \leq 1$ and $(y \geq x-1)$	вывод	верно?
А					
В					
С					
D					
E					
F					
G					
H					
J					
K					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).