

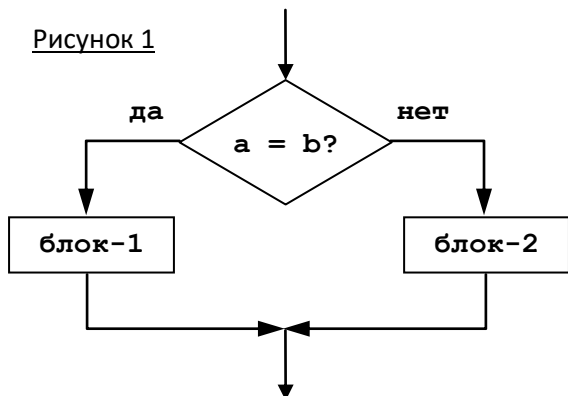
## 24 (С1) (повышенный уровень, время – 30 мин)

**Тема:** Исправление ошибок в простой программе с условными операторами.

**Что нужно знать:**

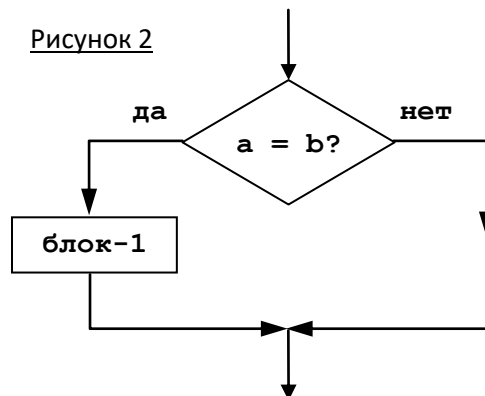
- правила построения программы на Паскале<sup>1</sup>, Python или Си
- правила работы с переменными (объявление, ввод, вывод, оператор присваивания)
- *ветвление* – это выбор одного из двух возможных вариантов действий в зависимости от того, выполняется ли некоторое условие;
- на блок-схеме алгоритма ветвление изображается в виде блока-ромба с одним входом и двумя выходами:

Рисунок 1



полная форма ветвления

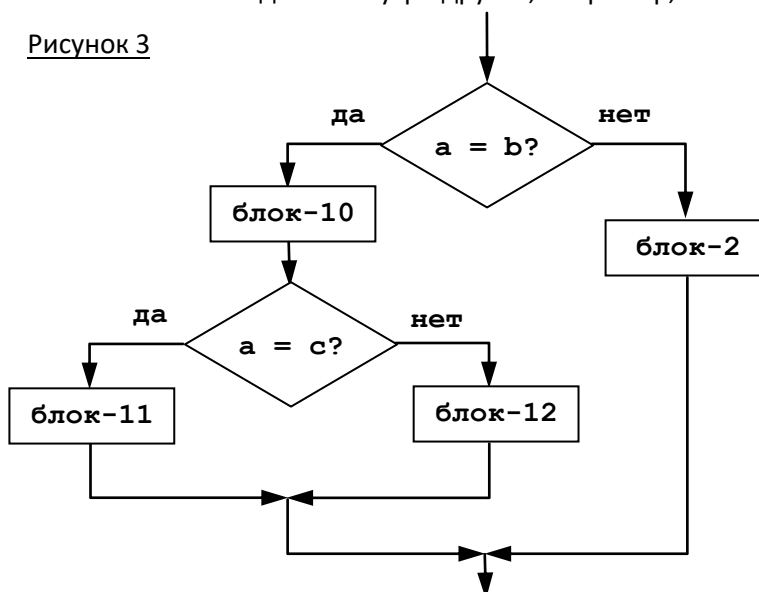
Рисунок 2



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 пустой, но это очень неграмотное решение, поскольку оно усложняет 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  $a \neq b$

<sup>1</sup> 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

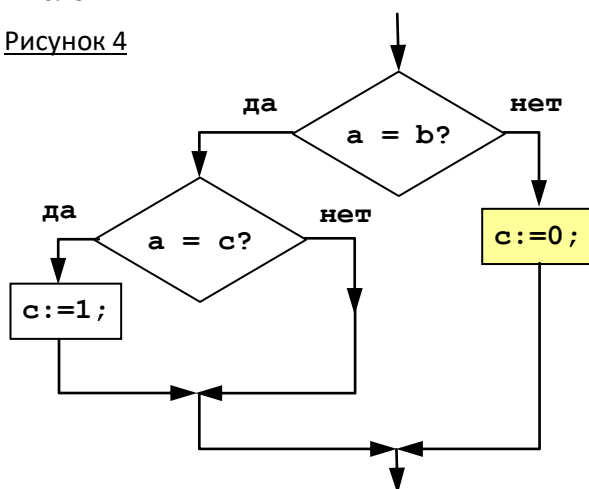
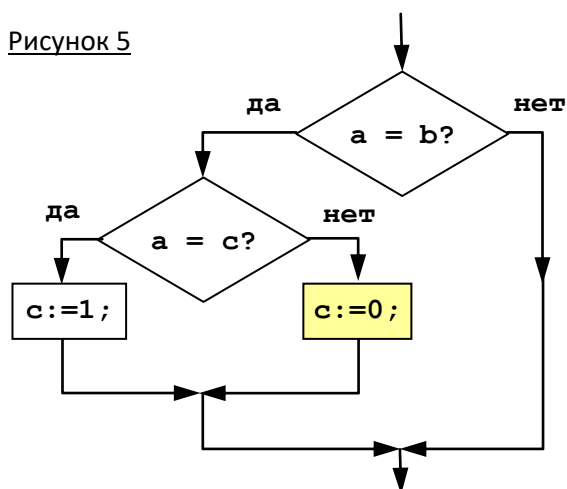


Рисунок 5



- 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:

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество нечётных чисел в исходной последовательности и максимальное нечётное число. Если нечётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Вот она:

```
const n = 4;
var i, x: integer;
var maximum, count: integer;
begin
    count := 0;
    maximum := 999;
    for i := 1 to n do begin
        read(x);
        if x mod 2 <> 0 then begin
            count := count + 1;
            if x > maximum then maximum := i
        end
    end;
    if count > 0 then begin
        writeln(count);
        writeln(maximum)
    end
    else writeln('NO')
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности: 2 9 4 3
2. Приведите пример такой последовательности, содержащей хотя бы одно нечётное число, что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

**Решение:**

- 1) обратим внимание на две строки в начале программы, которые начинаются с ключевого слова `var`: это не ошибка, такое повторение, действительно, допустимо в языке Паскаль; возможно, это была одна из ловушек разработчиков КИМ, которую они применили на реальном ЕГЭ-2014
- 2) теперь выполним программу для заданной последовательности 2 9 4 3, записывая все изменения переменных в таблицу:

	условие	i	x	maximum	count	вывод
<code>count:=0;</code>					0	
<code>maximum:=999;</code>				999		
<code>for i := 1 to n do</code>		1				
<code>  read(x);</code>			2			
<code>  if x mod 2 &lt;&gt; 0 then</code>	нет					
<code>end</code>		2				
<code>  read(x);</code>			9			
<code>  if x mod 2 &lt;&gt; 0 then</code>	да					
<code>    count:=count+1;</code>					1	
<code>    if x &gt; maximum then</code>	нет					
<code>end</code>		3				
<code>  read(x);</code>			4			
<code>  if x mod 2 &lt;&gt; 0 then</code>	нет					
<code>end</code>		4				
<code>  read(x);</code>			3			
<code>  if x mod 2 &lt;&gt; 0 then</code>	да					
<code>    count:=count+1;</code>					2	
<code>    if x &gt; maximum then</code>	нет					
<code>end</code>		5				
<code>if count &gt; 0 then</code>	да					
<code>  writeln(count);</code>						2
<code>  writeln(maximum)</code>						999

- 3) при ручной прокрутке программы мы увидели, что она правильно подсчитала количество нечётных чисел во входной последовательности, но неверно определила максимум: значение переменной `maximum`, которое было выведено на экран, осталось равным начальному значению 999, так как все остальные нечётные числа были меньше этого начального значения; поэтому ответ на п. 1 задания должен быть таким:

1) Программа выведет числа 2 и 999.

- 4) поскольку все числа по условию неотрицательны и не превышают 1000, программа всегда будет выдавать 999 вместо максимального нечётного числа; в то же время мы выяснили, что количество нечётных чисел в последовательности считается правильно; поэтому любая последовательность, содержащая 999, будет обрабатываться правильно
- 5) таким образом, правильный ответ на п. 2 должен быть таким:

2) Программа работает правильно для последовательности: 2 9 3 999.

- 6) теперь будем искать ошибки; как уже отмечалось, повторное использование ключевого слова `var` допустимо и указывать это в качестве ошибки нельзя!
- 7) как следует из результатов ручной прокрутки программы, во многих случаях она выдаёт неверный результат из-за того, что неверно задано начальное значение переменной `maximum`: оно должно быть **меньше**, чем любой возможный результат;

- 8) наименьшее нечётное неотрицательное число – это 1, поэтому можно принять в качестве начального значения `maximum` любое число, меньшее единицы (на самом деле, программа будет правильно работать и для 1), например:

3) **Ошибка 1.** `maximum:=999;`  
Исправление: `maximum:=0;`

- 9) если теперь (с исправленной первой ошибкой) сделать ручную прокрутку программы, то мы увидим, что на последовательности 2 9 4 3 она выдает сначала 2, а потом – 4, то есть, значение максимума вычисляется опять неверно;
- 10) откуда появится число 4 в переменной `maximum`? оно будет записана в результате выполнения оператора `if x > maximum then maximum:=i`, который записывает в переменную `maximum` не значение полученного числа (`x`), а его номер (`i`); таким образом, мы нашли вторую ошибку:

**Ошибка 2.** `if x > maximum then maximum:=i`  
Исправление: `if x > maximum then maximum:=x`

### Ещё пример задания:

Требовалось написать программу, которая вводит с клавиатуры натуральное число  $N$ , не превышающее  $10^9$ , и выводит сумму чётных цифр в десятичной записи этого числа или 0, если чётных цифр нет. Программист торопился и написал программу неправильно. Вот она:

```
var N: longint;
    s: integer;
begin
  readln(N);           (1)
  s := 0;              (2)
  while N > 1 do begin (3)
    if N mod 2 = 0 then begin (4)
      s := N mod 10;      (5)
    end;                 (6)
    N := N div 10;       (7)
  end;                   (8)
  write(s);             (9)
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1984.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

### Решение:

- 1) начнем с того, что разберёмся в условии задачи: ограничение  $N < 10^9$  введено для того, чтобы число поместилось в 4-байтовую переменную целого типа, для совместимости со всеми трансляторами Паскаля соответствующая переменная объявлена как `longint`
- 2) теперь разберём программу; строка (1) – это ввод исходного числа; очевидно, что переменная `s` – это результат, поскольку именно она выводится в строке (9)

- 3) в строке (2) переменная  $s$  обнуляется, это естественно при накоплении суммы
- 4) строки (3)-(8) – это цикл, который выполняется пока  $N > 1$ ; на каждом шаге цикла  $N$  делится на 10 нацело, то есть, из десятичной записи числа отбрасывается последняя цифра (строка (7)); такой цикл используется для того, чтобы перебрать все цифры числа, но обычно ставят условие « $N > 0$ », поскольку в приведенном варианте цикл остановится при  $N = 1$ ; однако в данном случае это **не влияет на результат**, поскольку по условию нас интересуют только чётные цифры, а 1 – нечётная
- 5) в строке (4) проверяется чётность числа (и одновременно чётность его последней цифры!), если число чётное, в строке (5) в переменную  $s$  записывается остаток от деления числа на 10, то есть последняя цифра десятичной записи этого числа
- 6) таким образом, после выполнения цикла будет выведена последняя рассмотренная цифра, для которой сработает условие в строке (4)
- 7) поскольку цифры перебираются с конца, выводится первая чётная цифра в записи числа
- 8) начнём выполнять задание:

1. При вводе числа 1984 будет выведено число 8 – значение первой чётной цифры числа.

- 9) когда программа выдаст верный ответ? очевидно, что тогда, когда сумма чётных цифр и значение первой чётной цифры совпадают; это возможно, если в числе
  - нет чётных цифр (сумма останется равной 0)
  - все чётные цифры – нули
  - одна чётная цифра
  - одна ненулевая чётная цифра, а нули стоят после неё (нули не меняют суммы!)

2. Программа выдаст правильный ответ для  $N = 1981$ .

- 10) как исправить программы? очевидно, нужно, чтобы она считала сумму чётных цифр, то есть, получив очередную чётную цифру, нужно добавить её к «старому» значению переменного  $s$ , изменив строку (5) так:

```
s := s + N mod 10; (5)
```

3. Ошибка допущена в строке

```
s := N mod 10;
```

Эта строка должна в правильной программе выглядеть так

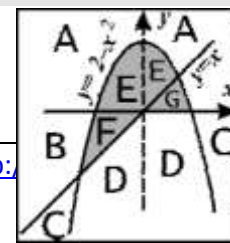
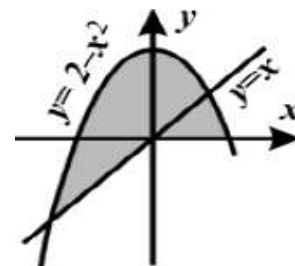
```
s := s + N mod 10;
```

### 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>=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  $y >= x$  истинно выше прямой  $y = x$ , то есть в областях A, B, E, F

Область	$y >= x$ ?	$y >= 0$ ?	$y <= 2 - x * x$ ?	вывод	верно?
A	да				
B	да				
C	нет				
D	нет				
E	да				
F	да				
G	нет				

- 3) условие  $y >= 0$  истинно выше прямой  $y = 0$ , то есть в областях A, E, G, однако это условие проверяется только тогда, когда первое условие,  $y >= x$ , 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	$y >= x$ ?	$y >= 0$ ?	$y <= 2 - x * x$ ?	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	$y >= x$ ?	$y >= 0$ ?	$y <= 2 - x * x$ ?	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	$y >= x$ ?	$y >= 0$ ?	$y <= 2 - x * x$ ?	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  $(y >= 0) \text{ and } (y <= 2 - x * x)$
- 9) область E+F соответствует условию  $(y >= x) \text{ and } (y <= 2 - x * x)$
- 10) объединение областей выполняется с помощью операции ИЛИ (or), так что полное условие принимает вид

$$(y >= 0) \text{ and } (y <= 2 - x * x) \text{ or } (y >= x) \text{ and } (y <= 2 - x * x)$$

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

$$((y >= 0) \text{ and } (y <= 2 - x * x)) \text{ or } ((y >= x) \text{ and } (y <= 2 - x * x))$$

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

$$(y <= 2 - x * x) \text{ and } ((y >= x) \text{ or } (y >= 0))$$

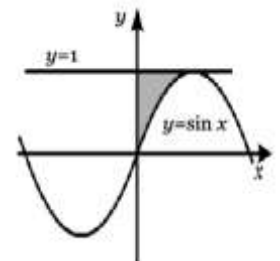
- 12) доработанная программа выглядит так:

```
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')
```



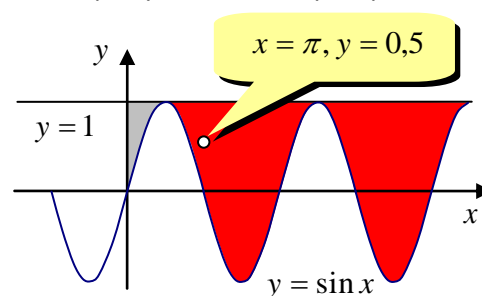
**end.**

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')

```

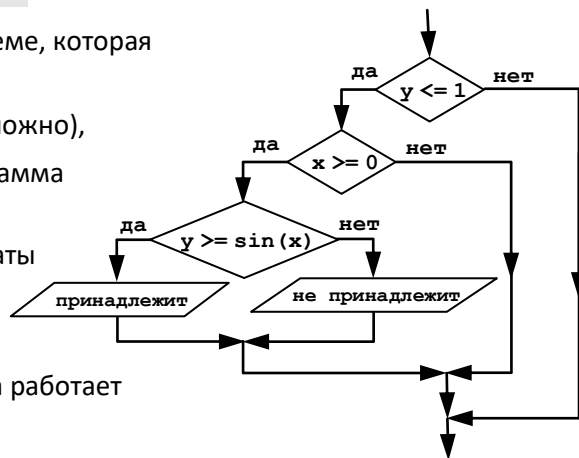
11) этот фрагмент программы соответствует блок-схеме, которая показана на рисунке справа:

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)

( $x=0$ ,  $y=2$ ) 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 – запятое, 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 тренировки<sup>2</sup>:

- 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

<sup>2</sup> Источники заданий:

1. Демонстрационные варианты ЕГЭ 2004-2013 гг.
2. Тренировочные и диагностические работы МИОО и Статград.
3. Гусева И.Ю. ЕГЭ. Информатика: раздаточный материал тренировочных тестов. — СПб: Тригон, 2009.
4. 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.
5. Абрамян М.Э., Михалкович С.С., Русанова Я.М., Чердынцева М.И. Информатика. ЕГЭ шаг за шагом. — М.: НИИ школьных технологий, 2010.
6. Крылов С.С., Ушаков Д.М. ЕГЭ 2015. Информатика. Тематические тестовые задания. — М.: Экзамен, 2015.
7. Ушаков Д.М. ЕГЭ-2015. Информатика. 20 типовых вариантов экзаменационных работ для подготовки к ЕГЭ. — М.: Астрель, 2014.

работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

- 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  $x$ ,  $y$ ,  $r1$ ,  $r2$ , 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

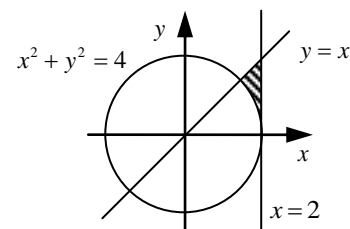
всех ненулевых действительных чисел  $a \neq 0$  и  $b \neq 0$ , введенных с клавиатуры. Программист торопился и сделал в программе ошибки.

```
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



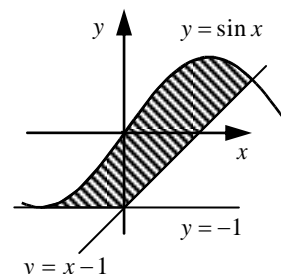
определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно. Вот она:

```
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.
```

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:

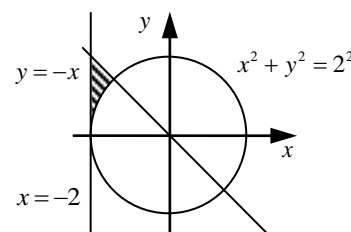
```
var x,y: real;
begin
  readln(x,y);
  if y >= -1 then
    if y <= sin(x) then
      if y >= x-1 then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
```



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.

```

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:

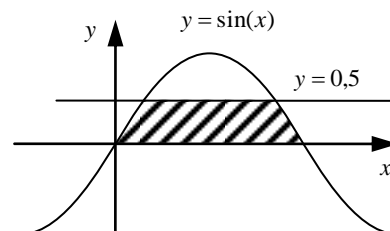
1. Приведите примеры таких чисел  $a$ ,  $b$ ,  $c$ , при которых программа неверно решает поставленную задачу.
2. Укажите, какая часть программы является лишней?
3. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).

- 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.

```



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:

```

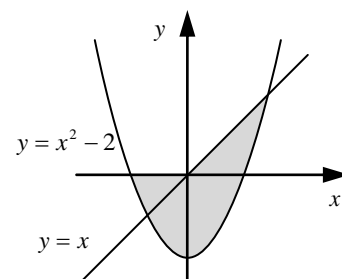
var x, y: real;

```

```

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

```



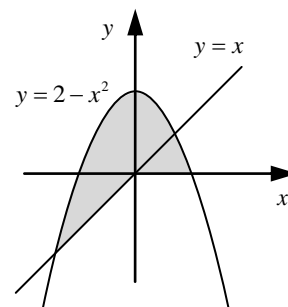
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой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

```



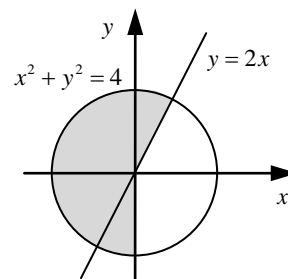
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой 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 >= 2*x then
    if x <= 0 then
      if x*x+y*y <= 4 then
        write('принадлежит')
      else
        write('не принадлежит');
      end.

```



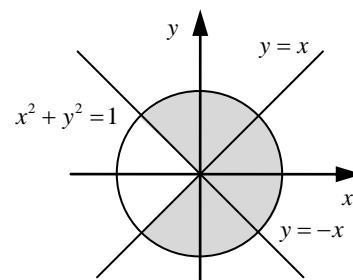
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой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

```



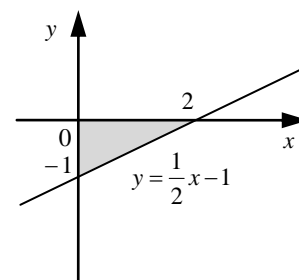
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой 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
  else
    write('не принадлежит');
  write('не принадлежит');
end.

```



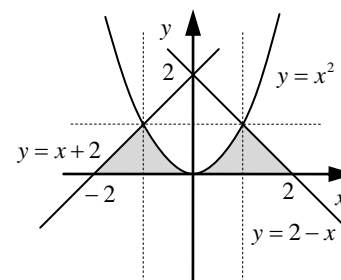
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x_0$ ,  $y_0$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой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

```

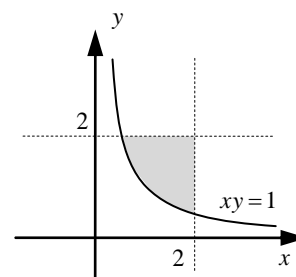
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой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

```



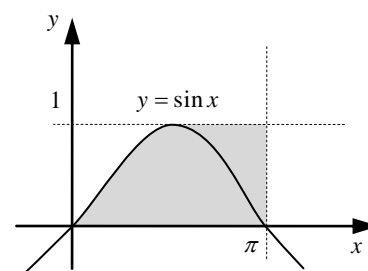
Последовательно выполните следующее: 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой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

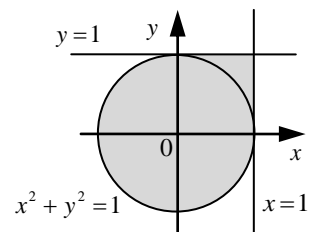
```



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.
```

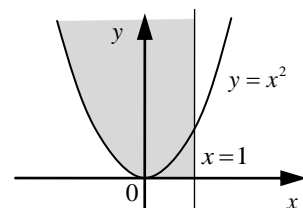


Последовательно выполните следующее:

- 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:

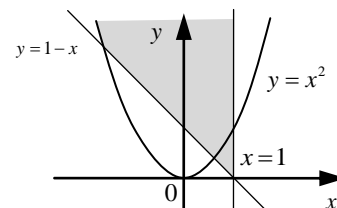
- 1) Приведите пример таких чисел  $x$ ,  $y$ , при которых программа неверно решает поставленную задачу. **Объясните, почему для указанных чисел программа неверно решает поставленную задачу.**



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.
```



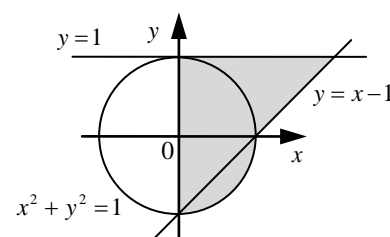
Последовательно выполните следующее:

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.
```

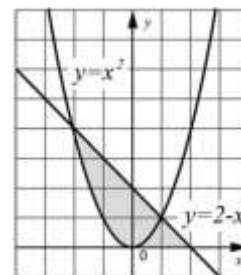


Последовательно выполните следующее:

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).

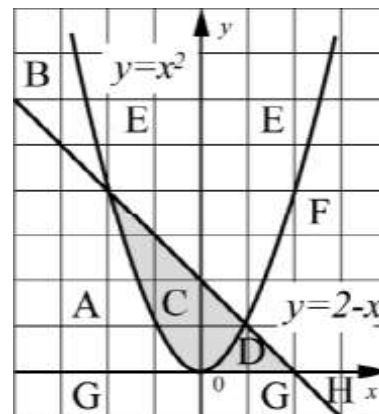
- 28) Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



```

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.
  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".

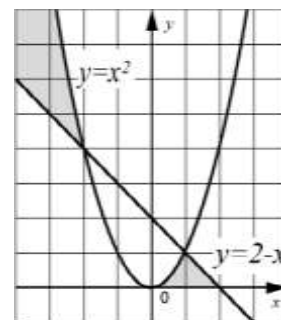
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 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 неправильно.

```

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

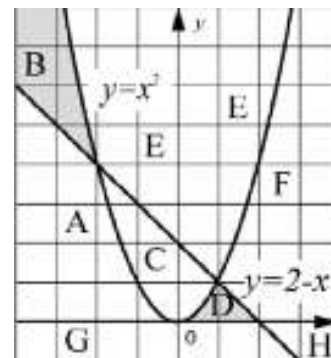
```



```

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".

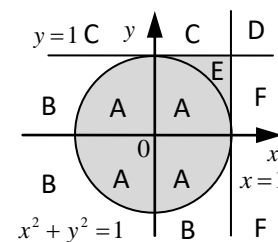
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 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?

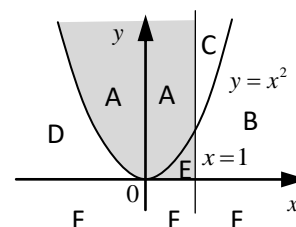
<b>A</b>					
<b>B</b>					
<b>C</b>					
<b>D</b>					
<b>E</b>					
<b>F</b>					

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.

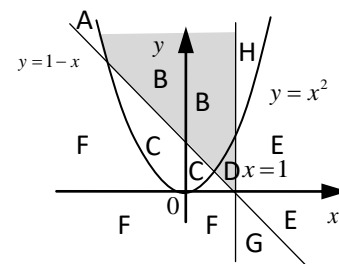
Область	$y \geq 0$	$x \leq 1$	$y \geq x*x$	вывод	верно?
<b>A</b>					
<b>B</b>					
<b>C</b>					
<b>D</b>					
<b>E</b>					
<b>F</b>					

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.
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.

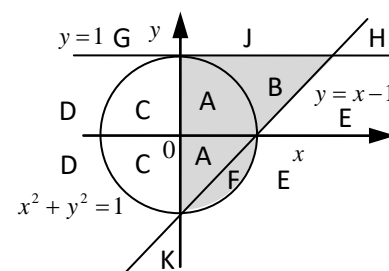
Область	$x \leq 1$	$y \geq 1-x$	$y \geq x^2$	вывод	верно?
A					
B					
C					
D					
E					
F					
G					
H					

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	$x \geq 0$	$y \leq 1$	$x*x + y*y \leq 1$ and $(y \geq x-1)$	вывод	верно?
А					
В					
С					
D					
Е					
F					
G					
Н					
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".

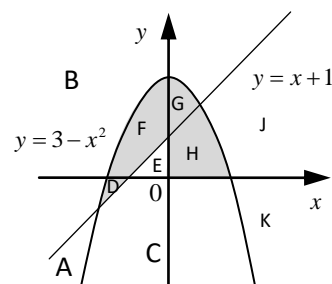
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

34) 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 <= 3-x*x then
if y >= 0 then
if 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). Точки, лежащие на границах областей, отдельно не рассматривать.

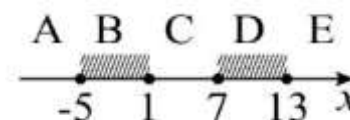
Область	$y \leq 3-x*x$	$y \geq 0$	$y \geq x+1$	вывод	верно?
А					
В					
С					

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).

- 35) Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков В и D (включая границы).



Программист торопился и написал программу неправильно.

```

var x: real;
begin
  readln(x) ;
  if x>1 then
    if x>=7 then
      if x>13 then
        write('не принадлежит')
      else
        write('принадлежит')
    end.

```

Последовательно выполните следующее.

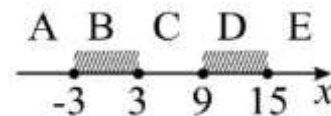
1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки -5, 1, 7 и 13) принадлежат заштрихованным областям.

Область	Условие 1 ( $x > 1$ )	Условие 2 ( $x \geq 7$ )	Условие 3 ( $x > 13$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

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 её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 36) Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков В и D (включая границы). Программист торопился и написал программу неправильно.



```

var x: real;
begin
  readln(x) ;
  if x<=15 then
    if x<9 then
      if x<=3 then
        write('принадлежит')
      else
        write('не принадлежит')
    end.

```

Последовательно выполните следующее.

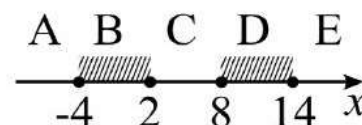
1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки -3, 3, 9 и 15) принадлежат заштрихованным областям.

Область	Условие 1 ( $x \leq 15$ )	Условие 2 ( $x < 9$ )	Условие 3 ( $x \leq 3$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

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 её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 37) Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков В и D (включая границы). Программист торопился и написал программу неправильно.



```

var x: real;
begin
  readln(x) ;
  if x<=14 then

```



```

if x>=-4 then
  if x>=8 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 и E). Границы (точки -4, 2, 8 и 14) принадлежат заштрихованным областям.

Область	Условие 1 ( $x \leq 14$ )	Условие 2 ( $x \geq -4$ )	Условие 3 ( $x \geq 8$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

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».

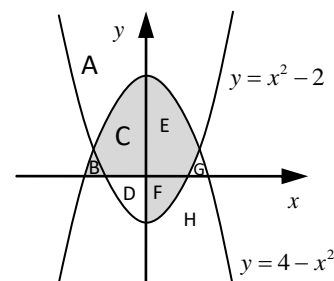
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

38) 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-2 then
    if y<=4-x*x then
      if x>=0 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 и E). Границы (точки -4, 2, 8 и 14) принадлежат заштрихованным областям.

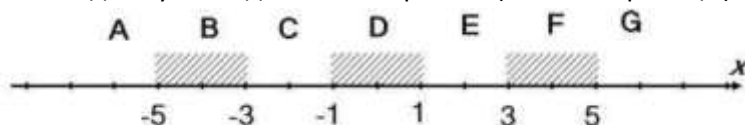
Область	Условие 1 ( $y \geq x^2 - 2$ )	Условие 2 ( $y \leq 4 - x^2$ )	Условие 3 ( $x \geq 0$ )	Программа выведет	Область обрабатывается верно
A					
B					

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 её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 39) Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  - действительное число) и определяется принадлежность этой точки одному из выделенных отрезков (включая границы).



Программист торопился и написал программу неправильно.

```
var x: real;
begin
  readln(x);
  if (x>=-5) or (x<=1) then
    if (x>=-3) and (x<=3) then
      if (x>=-1) and (x<=5) then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
end.
```

Последовательно выполните следующее.

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

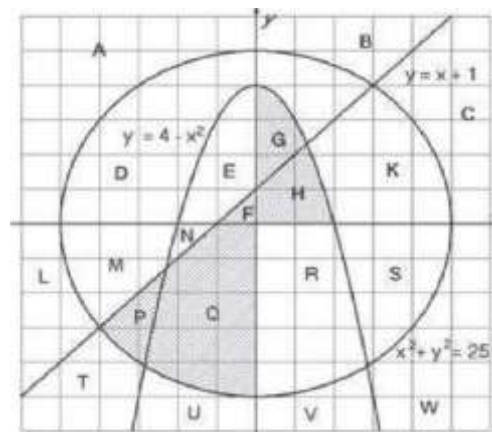
Область	Условие 1 ( $x \geq -5$ or $x \leq 1$ )	Условие 2 ( $x \geq -3$ and $x \leq 3$ )	Условие 3 ( $x \geq -1$ and $x \leq 5$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт 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 её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

- 40) 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).



Ученик написал такую программу:

```
var x, y: real;
begin
  readln(x, y);
  if x*x + y*y <= 25 then
    if y <= 4-x*x then
      if y <= x+1 then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
end.
```

При проверке работы программа выполнялась по шагам для некоторых контрольных значений  $x$  и  $y$ , при этом был заполнен протокол тестирования, содержащий следующую информацию.

**Область** - часть плоскости, которой принадлежит проверяемая точка (все возможные области отмечены на рисунке буквами А, В, С, ... W).

**Условие 1, Условие 2, Условие 3** - результат проверки соответствующего условия (да или нет). Если условие не проверялось, в протокол записывался прочерк.

**Вывод** - сообщение, которое вывела программа. Если программа ничего не вывела, в протокол записывался прочерк.

**Верно** - итоговое заключение (да или нет) о правильности результата работы программы при данных значениях  $x$  и  $y$ .

В результате неаккуратного обращения протокол был испорчен, частично сохранились только три строки:

Область	Условие 1 ( $x*x + y*y \leq 25$ )	Условие 2 ( $y \leq 4-x*x$ )	Условие 3 ( $y \leq x+1$ )	Вывод	Верно
	да			—	
				принадлежит	нет
			нет		нет

Последовательно выполните следующее.

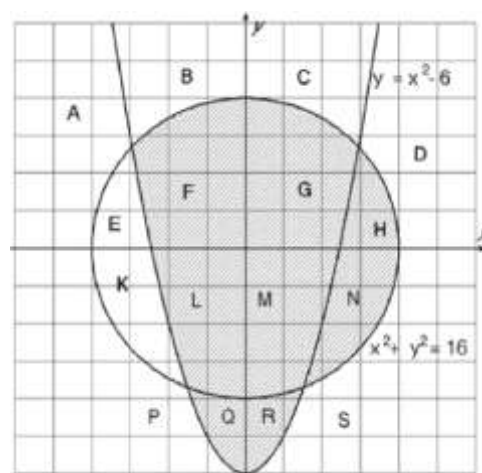
1. Восстановите уцелевшие строки протокола, заполнив все клетки таблицы. Там, где содержание восстанавливается неоднозначно, запишите любое возможное значение.

Например, если для нескольких областей получается одинаковая строка таблицы, укажите в графе «Область» любую из этих областей.

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

- 41) 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). Ученик написал такую программу:

```
var x, y: real;
begin
  readln(x,y);
  if y >= x*x-6 then begin
    if x*x + y*y <=16 then
      write('принадлежит')
    end
  else
    if x >= 0 then
      write('принадлежит')
    else write('не принадлежит')
  end.
end.
```



При проверке работы программа выполнялась по шагам для некоторых контрольных значений  $x$  и  $y$ , при этом был заполнен протокол тестирования. В результате неаккуратного обращения протокол был испорчен, частично сохранились только четыре строки:

Область	Условие 1 ( $y \geq x^2 - 6$ )	Условие 2 ( $x^2 + y^2 \leq 16$ )	Условие 3 ( $x \geq 0$ )	Вывод	Верно
	да			—	
				принадлежит	нет
				не принадлежит	да
			—		да

Графы протокола содержат следующую информацию.

**Область** - часть плоскости, которой принадлежит проверяемая точка. (Все возможные области отмечены на рисунке буквами А, В, С, ... S.)

**Условие 1, Условие 2, Условие 3** - результат проверки соответствующего условия (да или нет). Если условие не проверялось, в протокол записывался прочерк.

**Вывод** – сообщение, которое вывела программа. Если программа ничего не вывела, в протокол записывался прочерк.

**Верно** - итоговое заключение (да или нет) о правильности результата работы программы при данных значениях  $x$  и  $y$ .

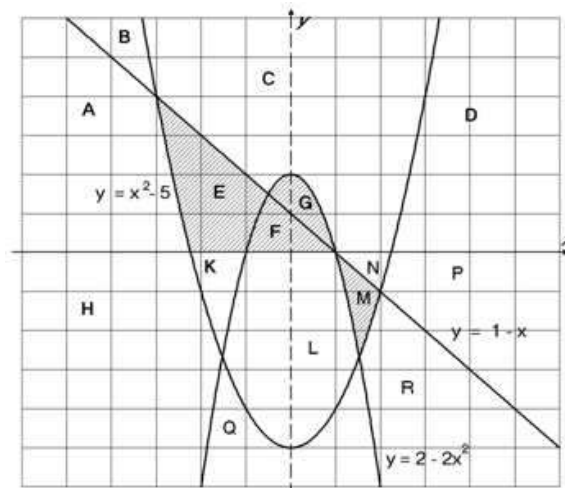
Последовательно выполните следующее.

- Восстановите уцелевшие строки протокола, заполнив все клетки таблицы. Там, где содержание восстанавливается неоднозначно, запишите любое возможное значение. Например, если для нескольких областей получается одинаковая строка таблицы, укажите в графе «Область» любую из этих областей.
- Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

42) 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).

Ученик написал такую программу:

```
var x, y: real;
begin
  readln(x,y);
  if y >= 1-x then begin
    if y <= 2-2*x*x then
      write('принадлежит')
    end
  else
    if y >= x*x-5 then
      write('принадлежит')
    else write('не принадлежит')
  end.
end.
```



При проверке работы программа выполнялась по шагам для некоторых контрольных значений  $x$  и  $y$ , при этом был заполнен протокол тестирования. В результате неаккуратного обращения протокол был испорчен, частично сохранились только четыре строки:

Область	Условие 1 ( $y \geq 1-x$ )	Условие 2 ( $y \leq 2-2*x*x$ )	Условие 3 ( $y \geq x*x-5$ )	Вывод	Верно
				—	
				принадлежит	нет
				не принадлежит	да
			—		да

Графы протокола содержат следующую информацию.

**Область** - часть плоскости, которой принадлежит проверяемая точка. (Все возможные области отмечены на рисунке буквами А, В, С, ... R.)

**Условие 1, Условие 2, Условие 3** - результат проверки соответствующего условия (да или нет). Если условие не проверялось, в протокол записывался прочерк.

**Вывод** – сообщение, которое вывела программа. Если программа ничего не вывела, в протокол записывался прочерк.

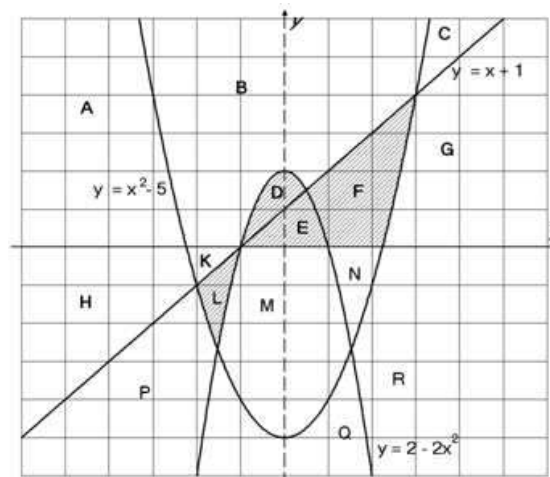
**Верно** - итоговое заключение (да или нет) о правильности результата работы программы при данных значениях  $x$  и  $y$ .

Последовательно выполните следующее.

1. Восстановите уцелевшие строки протокола, заполнив все клетки таблицы. Там, где содержание восстанавливается неоднозначно, запишите любое возможное значение. Например, если для нескольких областей получается одинаковая строка таблицы, укажите в графе «Область» любую из этих областей.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

43) 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). Ученик написал такую программу:

```
var x, y: real;
begin
  readln(x,y);
  if y >= x+1 then begin
    if y <= 2-2*x*x then
      write('принадлежит')
    end
  else
    if y >= x*x-5 then
      write('принадлежит')
    else write('не принадлежит')
  end.
end.
```



При проверке работы программа выполнялась по шагам для некоторых контрольных значений  $x$  и  $y$ , при этом был заполнен протокол тестирования. В результате неаккуратного обращения протокол был испорчен, частично сохранились только четыре строки:

Область	Условие 1 ( $y \geq x+1$ )	Условие 2 ( $y \leq 2-2*x*x$ )	Условие 3 ( $y \geq x*x-5$ )	Вывод	Верно
				—	
				принадлежит	нет
				не принадлежит	да
			—		да

Графы протокола содержат следующую информацию.

**Область** - часть плоскости, которой принадлежит проверяемая точка. (Все возможные области отмечены на рисунке буквами А, В, С, ... R.)

**Условие 1, Условие 2, Условие 3** - результат проверки соответствующего условия (да или нет). Если условие не проверялось, в протокол записывался прочерк.

**Вывод** – сообщение, которое вывела программа. Если программа ничего не вывела, в протокол записывался прочерк.

**Верно** - итоговое заключение (да или нет) о правильности результата работы программы при данных значениях  $x$  и  $y$ .

Последовательно выполните следующее.

1. Восстановите уцелевшие строки протокола, заполнив все клетки таблицы. Там, где содержание восстанавливается неоднозначно, запишите любое возможное значение. Например, если для нескольких областей получается одинаковая строка таблицы, укажите в графе «Область» любую из этих областей.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

44) В корзине лежит  $a$  красных и  $b$  синих шаров. Из корзины наугад вынимают  $n$  шаров. Требовалось написать программу, которая определяет количество вариантов такого опыта (вариантом считаем пару чисел: количество вынутых красных и синих шаров). Программист торопился и написал программу неправильно:

Pascal	Python
<pre>var n, a, b: integer; begin   readln(n, a, b);   if a + b &lt; n then writeln(0)   else     if a &lt; b then       if a &lt; n then writeln(a+1)     else       if b &lt; n then writeln(b+1)     else writeln(n+1); end.</pre>	<pre>n, a, b = map(int, input().split()) if a + b &lt; n:   print(0) else:   if a &lt; b:     if a &lt; n:       print(a+1)   else:     if b &lt; n:       print(b+1)   else:     print(n+1)</pre>

Последовательно выполните следующее:

1. Определите, что выведет программа при  $n = 5$ ,  $a = 3$ ,  $b = 6$ .
2. Укажите значения  $n$ ,  $a$ ,  $b$ , при которых программа выдаёт неправильный ответ.
3. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы.

- 45) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится произведение цифр этого числа.

Программист торопился и написал программу неправильно:

Pascal	Python
<pre>var N, product: longint;     digit: integer; begin   readln(N);   product := N mod 10;   while N &gt;= 10 do begin     digit := N mod 10;     product := product * digit;     N := N div 10   end;   writeln ( product ) end.</pre>	<pre>N = int(input()) product = N % 10 while N &gt;= 10:   digit = N % 10   product = product * digit   N = N // 10 print(product)</pre>

Последовательно выполните следующее:

1. Определите, что выведет программа при вводе числа 532.
2. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - а) выпишите строку, в которой сделана ошибка;
  - б) укажите, как надо исправить ошибку - приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, имеющую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

- 46) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится минимальную цифру этого числа.

Программист торопился и написал программу неправильно:

Pascal	Python
<pre>var N: longint;     digit, min_digit: integer; begin</pre>	<pre>N = int(input()) min_digit = 0 while N &gt; 0:</pre>



<pre> readln(N) ; min_digit := 0; while N &gt; 0 do begin     digit := N mod 10;     if digit &gt; min_digit then         min_digit := digit;     N := N div 10; end; writeln(min_digit) ; end. </pre>	<pre> digit = N % 10 if digit &gt; min_digit:     min_digit = digit N = N // 10 print(min_digit) </pre>
--	---

Последовательно выполните следующее:

1. Определите, что выведет программа при вводе числа 532.
2. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - а) выпишите строку, в которой сделана ошибка;
  - б) укажите, как надо исправить ошибку - приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, имеющую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

- 47) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится количество цифр этого числа.

Программист торопился и написал программу неправильно:

Pascal	Python
<pre> var N: longint;     count: integer; begin     readln(N) ;     count := 1;     while N &gt; 1 do begin         count := count + 1;         N := N div 10;     end;     writeln(count) ; end. </pre>	<pre> N = int(input()) count = 1 while N &gt; 1:     count = count + 1     N = N // 10 print(count) </pre>

Последовательно выполните следующее:

1. Определите, что выведет программа при вводе числа 123.
2. Укажите одно число, для которого программа выводит неверный результат.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - а) выпишите строку, в которой сделана ошибка;
  - б) укажите, как надо исправить ошибку - приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, имеющую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

- 48) Требовалось написать программу, при выполнении которой с клавиатуры вводится натуральное число, не превосходящее  $10^8$ , и выводится его первая (старшая) цифра. Ученик написал такую программу:

Pascal	Python
<pre> var n: longint; begin </pre>	<pre> n = int(input()) while n &gt; 10: </pre>



```

read(n) ;
while n>10 do begin
  n := n mod 10
end;
write(n) ;
end.

```

```

n = n % 10
print(n)

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1984.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

- 49) Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, которое получается из  $N$  после удаления всех единиц; порядок остальных цифр при этом не меняется. Например, число 19520125 должно быть преобразовано в число 952025. Число, в котором все цифры – единицы и нули, должно быть преобразовано в 0. Незначащие нули в старших разрядах полученного числа печатать не нужно. Программист торопился и написал программу неправильно.

Pascal	Python
<pre> var N, R, T: longint;   d: integer; begin   readln(N) ;   R:=0;   T:=1;   while N&gt;0 do begin     d := N mod 10;     if d&lt;&gt;1 then begin       R := R + d*T;       T := T+1     end;     N := N div 10;   end;   writeln(T) ; end. </pre>	<pre> N = int(input()) R = 0 T = 1 while N &gt; 0:     d = N % 10     if d != 1:         R = R + d*T         T = T + 1     N = N // 10 print(T) </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 314.
2. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - 1) Выпишите строку, в которой сделана ошибка.
  - 2) Укажите, как исправить ошибку – приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

- 50) Дано натуральное число  $N$ , не превосходящее  $10^8$ . Необходимо найти и вывести число, которое получится при записи  $N$  справа налево и удалении всех единиц. Ведущие нули выводить не надо.

Если в числе N нет цифр кроме единиц и нулей, необходимо вывести 0. Например, при вводе числа 1984 нужно вывести 489, а при вводе 2001 нужно вывести 2. Для решения этой задачи ученик написал такую программу:

Pascal	Python
<pre> var n, m: longint; d: integer; begin   read(n);   m := 0;   while n &gt;= 1 do begin     d := n mod 10;     if d &gt; 1 then begin       m := 10*d + m;     end;     n := (n - d) div 10;   end;   write(m); end. </pre>	<pre> n = int(input()) m = 0 while n &gt;= 1:     d = n % 10     if d &gt; 1:         m = 10*d + m     n = (n - d) // 10 print(m) </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1984.
2. Приведите пример числа, при вводе которого программа выдаст верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки: выпишите строку, в которой сделана ошибка, и приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

51) На вход программы поступают 4 неотрицательных целых числа, не превышающие 1000, среди которых могут быть одинаковые. Нужно написать программу, которая выводит количество чисел, кратных 3, и максимальное из этих чисел. Если среди входных данных нет чисел, кратных трём, программа должна вывести слово 'NO'. Для решения этой задачи ученик написал такую программу:

Pascal	Python
<pre> const n = 4; var i, x: integer; var maximum, count: integer; begin   count := 0;   maximum := 999;   for i:=1 to n do begin     read( x );     if x mod 3 = 0 then begin       count := count + 1;       if x &lt; maximum then         maximum := x;     end;   end;   if count &gt; 0 then begin     writeln(count);     writeln(maximum);   end; end. </pre>	<pre> n = 4 count = 0 maximum = 999 for i in range(1, n+1):     x = int(input())     if x % 3 == 0:         count = count + 1         if x &lt; maximum:             maximum = x if count &gt; 0:     print(count)     print(maximum) else:     print('NO') </pre>

<pre> end else     writeln('NO'); end.</pre>	
--	--

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 2 9 4 3.
2. Приведите пример такой последовательности, содержащей число, кратное 3, что, несмотря на ошибки, программа выдаст верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки: выпишите строку, в которой сделана ошибка, и приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

52) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится сумма цифр этого числа. Программист торопился и написал программу неправильно:

Pascal	Python
<pre> var N: longint;     sum, d: integer; begin     readln(N);     sum := 1;     while N &gt; 0 do begin         d := N mod 10;         N := N div 10;         sum := sum + 1;     end;     writeln(sum); end.</pre>	<pre> N = int(input()) sum = 1 while N &gt; 0:     d = N % 10     N = N // 10     sum = sum + 1 print(sum)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 256.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт правильный результат.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т. е. приведите правильный вариант строки.

53) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится сумма цифр этого числа. Программист торопился и написал программу неправильно:

Pascal	Python
<pre> var N: longint;     sum, d: integer; begin     readln(N);     sum := 1;     while N &gt; 0 do begin         d := N mod 10;         N := N div 10;</pre>	<pre> N = int(input()) sum = 1 while N &gt; 0:     d = N % 10     N = N // 10     sum = d print(sum)</pre>

<pre> sum := d; end; writeln(sum); end. </pre>	
--	--

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 256.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт правильный результат.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т. е. приведите правильный вариант строки.

54) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $x$ , не превосходящее 1000, и выводится количество значащих цифр в двоичной записи этого числа. Программист торопился и написал программу неправильно.

Pascal	Python
<pre> var x,cnt: integer; begin   readln(x);   cnt := 0;   while x &gt; 0 do begin     cnt:=cnt + x mod 2;     x := x div 10   end;   writeln(cnt) end. </pre>	<pre> x = int(input()) cnt = 0 while x &gt; 0:     cnt = cnt + x % 2     x = x // 10 print(cnt) </pre>

Последовательно выполните следующее:

1. Напишите, что выведет эта программа при вводе числа 15.
2. Приведите пример такого числа  $x$ , что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

55) Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $x$ , не превосходящее 1000, и выводится количество единиц в двоичной записи этого числа. Программист торопился и написал программу неправильно.

Pascal	Python
<pre> var x,cnt: integer; begin   readln(x);   cnt := 0;   while x &gt; 0 do begin     cnt:=cnt + 1;     x := x mod 2   end;   writeln(cnt) end. </pre>	<pre> x = int(input()) cnt = 0 while x &gt; 0:     cnt = cnt + 1     x = x % 2 print(cnt) </pre>

Последовательно выполните следующее:

1. Напишите, что выведет эта программа при вводе числа 6.
2. Приведите пример такого числа  $x$ , что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

56) Для заданного положительного вещественного числа  $A$  необходимо найти минимальное целое число  $K$ , при котором выполняется неравенство  $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{K} \geq A$ . Для решения этой задачи ученик написал такую программу:

Pascal	Python
<pre>var a, s: real;     k: integer; begin   read(a);   k := 1;   s := 1;   while s&lt;=a do begin     k := k + 1;     s := 1.0/k;   end;   write(k); end.</pre>	<pre>a = int(input()) k = 1 s = 1 while s &lt;= a:     k = k + 1     s = 1.0 / k print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 0.4.
2. Сколько существует натуральных чисел  $A$ , при вводе которых программа выведет ответ 2?
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

57) Для заданного положительного вещественного числа  $A$  необходимо найти минимальное целое число  $K$ , при котором выполняется неравенство  $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{K} \geq A$ . Для решения этой задачи ученик написал такую программу:

Pascal	Python
<pre>var a, s: real;     k: integer; begin   read(a);   k := 0;   s := 1;   while s &gt;= a do begin     k := k + 1;     s := s + 1.0/k;   end;   write(k); end.</pre>	<pre>a = float(input()) k = 0 s = 1 while s &gt;= a:     k = k + 1     s = s + 1.0/k print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1.4.

2. Сколько существует натуральных чисел А, при вводе которых программа выведет ответ 1?  
 3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

58) На обработку поступает последовательность из четырёх целых чисел. Нужно написать программу, которая выводит на экран количество неотрицательных чисел последовательности и их произведение. Если неотрицательных чисел нет, требуется вывести на экран «NO». Известно, что вводимые числа по абсолютной величине не превышают 10. Программист написал программу неправильно.

Pascal	Python
<pre> var p,i,x,count: integer; begin   count := 0;   p := 0;   for i := 1 to 4 do begin     read (x);     if x &gt;= 0 then begin       p := p*x;       count := count+1     end   end;   if count &gt; 0 then begin     writeln(x);     writeln(p);   end else   writeln('NO') end. </pre>	<pre> count = 0 p = 0 for i in {1, 2, 3, 4}:   x = int(input())   if x &gt;= 0:     p = p * x     count = count + 1 if count &gt; 0:   print(x)   print(p) else:   print('NO') </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел -5 2 1 3.
2. Приведите пример такой последовательности, содержащей хотя бы одно неотрицательное число, что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

59) На обработку поступает последовательность из четырёх целых чисел. Нужно написать программу, которая выводит на экран сумму всех отрицательных чисел последовательности и максимальное число в последовательности. Известно, что вводимые числа по абсолютной величине не превышают 1000. Программист написал программу неправильно.

Pascal	Python
<pre> var s,i,x,mx: integer; begin   mx := 0;   s := 0;   for i := 1 to 4 do begin     read (x);     if x &lt; 0 then       s := x;     if x &gt; mx then       mx := x;   end;   writeln(s); </pre>	<pre> mx = 0 s = 0 for i in {1, 2, 3, 4}:   x = int(input())   if x &lt; 0:     s = x   if x &gt; mx:     mx = x print(s) print(mx) </pre>

<code>writeln(mx)</code> <code>end.</code>	
---	--

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел `-5 2 -4 3`.
2. Приведите пример такой последовательности, содержащей хотя бы одно неотрицательное число, что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

- 60) На обработку поступает натуральное число, не превышающее  $10^9$ . Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 3. Если в числе нет цифр, кратных 3, требуется на экран вывести «NO». Программист написал программу неправильно.  
Напоминание: 0 делится на любое натуральное число.

Pascal	Python
<pre>var N,digit,maxDigit: longint; begin readln(N); maxDigit := N mod 10; while N &gt; 0 do begin   digit := N mod 10;   if digit mod 3 = 0 then     if digit &gt; maxDigit then       maxDigit := digit;   N := N div 10; end; if maxDigit = 0 then   writeln('NO') else   writeln(maxDigit) end.</pre>	<pre>N = int(input()) maxDigit = N % 10 while N &gt; 0:   digit = N % 10   if digit % 3 == 0:     if digit &gt; maxDigit:       maxDigit = digit   N = N // 10 if maxDigit == 0:   print('NO') else:   print(maxDigit)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа `134`.
2. Приведите пример такого трёхзначного числа, что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

- 61) Для заданного положительного вещественного числа  $A$  необходимо найти максимальное целое число  $K$ , при котором выполняется неравенство  $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{K} < A$ . Для решения этой задачи ученик написал такую программу:

Pascal	Python
<pre>var a, s: real;     k: integer; begin   read(a);   k := 0;   s := 1;   while s &lt; a do begin     k := k + 1;     s := s + 1.0/k;   end;</pre>	<pre>a = float(input()) k = 0 s = 1 while s &lt; a:   k = k + 1   s = s + 1.0/k print(k)</pre>

<code>write(k);</code> <code>end.</code>	
---	--

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1.8.
2. Приведите пример числа, при вводе которого программа выведет правильный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

62) Для заданного положительного вещественного числа  $A$  необходимо найти максимальное целое число  $K$ , при котором выполняется неравенство  $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{K} < A$ . Для решения этой задачи ученик написал такую программу:

Pascal	Python
<pre>var a, s: real;     k: integer; begin   read(a);   k := 1;   s := 1;   while s &lt; a do begin     s := s + 1.0/k;     k := k + 1;   end;   write(k); end.</pre>	<pre>a = float(input()) k = 1 s = 1 while s &lt; a:     s = s + 1.0/k     k = k + 1 print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1.2.
2. Приведите пример числа, при вводе которого программа выведет правильный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

63) На обработку поступает положительное целое число, не превышающее  $10^9$ . Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно.

Pascal	Python
<pre>var N, digit, sum: longint; begin   readln(N);   sum := 0;   while N &gt; 0 do begin     digit := N mod 10;     if digit &lt; 7 then       sum := sum + 1;     N := N div 10;   end;   writeln(digit) end.</pre>	<pre>N = int(input()) sum = 0 while N &gt; 0:     digit = N % 10     if digit &lt; 7:         sum = sum + 1     N = N // 10 print(digit)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 456.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько).



64) Дано целое положительное число  $N$ . Необходимо определить наименьшее целое число  $K$ , для которого выполняется неравенство:

$$1 + 2 + \dots + K > N.$$

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre>var n, k: integer; begin   read(n);   k := 1;   while n &gt; 0 do begin     n := n - k;     k := k + 1;   end;   writeln(k) end.</pre>	<pre>n = int(input()) k = 1 while n &gt; 0:     n = n - k     k = k + 1 print(k)</pre>

Последовательно выполните следующее.

1. Приведите пример числа  $N$ , при вводе которого программа выведет неверный ответ. Укажите верный ответ и ответ, который выведет программа.
2. Приведите пример числа  $N$ , при вводе которого программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

65) Дано целое положительное число  $N$ . Необходимо определить наименьшее целое число  $K$ , для которого выполняется неравенство:

$$1 + 2 + \dots + K \geq N.$$

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre>var n, k: integer; begin   read(n);   k := 1;   while n &gt;= 0 do begin     k := k + 1;     n := n - k;   end;   writeln(k) end.</pre>	<pre>n = int(input()) k = 1 while n &gt;= 0:     k = k + 1     n = n - k print(k)</pre>

Последовательно выполните следующее.

1. Приведите пример числа  $N$ , при вводе которого программа выведет неверный ответ. Укажите верный ответ и ответ, который выведет программа.
2. Приведите пример числа  $N$ , при вводе которого программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

66) Даны целые положительные числа  $M$  и  $N$ . Необходимо определить количество таких целых чисел  $K$ , для которых выполняется неравенство  $M \leq K^3 \leq N$ . Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre>var m, n, k, t: integer;</pre>	<pre>m, n = map(int, input().split())</pre>

<pre>begin   read(m, n);   k := 1;   t := 1;   while k*k*k &lt;= n do begin     if k*k*k &gt; m then t := t + 1;     k := k + 1;   end;   writeln(t) end.</pre>	<pre>k = 1 t = 1 while k*k*k &lt;= n:   if k*k*k &gt; m:     t = t + 1   k = k + 1 print(t)</pre>
---	---

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе  $M = 10$  и  $N = 100$ .
2. Приведите пример таких чисел  $M$  и  $N$ , при вводе которых программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

67) Даны целые положительные числа  $M$  и  $N$ . Необходимо определить количество таких натуральных чисел  $K$ , для которых выполняется неравенство  $M \leq K^2 \leq N$ . Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre>var m, n, k, t: integer; begin   read(m, n);   k := 1;   t := 1;   while k*k &lt; n do begin     if k*k &gt;= m then t := t + 1;     k := k + 1;   end;   writeln(t) end.</pre>	<pre>m, n = map(int, input().split()) k = 1 t = 1 while k*k &lt; n:   if k*k &gt;= m:     t = t + 1   k = k + 1 print(t)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе  $M = 10$  и  $N = 40$ .
2. Приведите пример таких чисел  $M$  и  $N$ , при вводе которых программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

68) Дано целое положительное число  $N$ , не превосходящее 1000. Необходимо определить, является ли это число степенью числа 5. То есть требуется определить, существует ли такое целое число  $K$ , что  $5^K = N$ , и вывести это число либо сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

Pascal	Python
<pre>var n, k: integer; begin   read(n);   k := 0;   while k mod 5 = 0 do begin     k := k + 1;     n := n div 5;   end;</pre>	<pre>n = int(input()) k = 0 while k % 5 == 0:   k = k + 1   n = n // 5 if k == 1:   print(k) else:</pre>

<pre> if k = 1 then   writeln(k) else   writeln('Не существует') end. </pre>	<pre> print('Не существует') </pre>
--	-------------------------------------

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 25.
  2. Приведите пример числа, при вводе которого приведённая программа напечатает то, что требуется.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 69) Дано целое положительное число N, не превосходящее 1000. Необходимо определить, является ли это число степенью числа 7. То есть требуется определить, существует ли такое целое число K, что  $7^K = N$ , и вывести это число либо сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

Pascal	Python
<pre> var n, k: integer; begin   read(n);   k := 0;   while n mod 7 = 0 do begin     k := k + n div 7;     n := n div 7;   end;   if n &lt;= 7 then     writeln(k)   else     writeln('Не существует') end. </pre>	<pre> n = int(input()) k = 0 while n % 7 == 0:     k = k + n // 7     n = n // 7 if n &lt;= 7:     print(k) else:     print('Не существует') </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 49.
  2. Приведите пример числа, при вводе которого приведённая программа напечатает то, что требуется.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 70) Дано целое положительное число N, не превосходящее 1000. Необходимо определить, является ли это число степенью числа 4. То есть требуется определить, существует ли такое целое число K, что  $4^K = N$ , и вывести это число либо сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

Pascal	Python
<pre> var n, k: integer; begin   read(n);   k := 0;   while n mod 4 = 0 do begin     k := k + n div 4;     n := n div 4;   end;   if k = 1 then     writeln(k) </pre>	<pre> n = int(input()) k = 0 while n % 4 == 0:     k = k + n // 4     n = n // 4 if k == 1:     print(k) else:     print('Не существует') </pre>

<pre> else     writeln('Не существует') end.</pre>	
--	--

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 16.
  2. Приведите пример числа, при вводе которого приведённая программа напечатает то, что требуется.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 71) Дано целое положительное число N, не превосходящее 1000. Необходимо определить, является ли это число степенью числа 3. То есть требуется определить, существует ли такое целое число K, что  $3^K = N$ , и вывести это число либо сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

Pascal	Python
<pre> var n, k: integer; begin     read(n);     k := 0;     while k mod 3 = 0 do begin         k := k + 1;         n := n div 3     end;     if n &gt; 0 then         writeln(k)     else         writeln('Не существует')     end.</pre>	<pre> n = int(input()) k = 0 while k % 3 == 0:     k = k + 1     n = n // 3 if n &gt; 0:     print(k) else:     print('Не существует')</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 9.
  2. Приведите пример числа, при вводе которого приведённая программа напечатает то, что требуется.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 72) Дано целое положительное число A. Требуется вывести такое минимально возможное нечётное натуральное число K, при котором сумма квадратов первых нечётных чисел  $1^2 + 3^2 + \dots + K^2$  окажется больше A. Для решения этой задачи ученик написал программу, но, к сожалению, его программа – неправильная.

Pascal	Python
<pre> var a, s, k: integer; begin     read(A);     s := 0;     k := -1;     while s &lt; A do begin         k := k + 1;         s := s + k*k;     end;     writeln(k) end.</pre>	<pre> A = int(input()) s = 0 k = -1 while s &lt; A:     k = k + 1     s = s + k*k print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 11.
  2. Укажите наименьшее значение А, при котором программа выведет верный ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 73) Дано целое положительное число А. Требуется вывести такое минимально возможное нечётное натуральное число К, при котором сумма квадратов первых нечётных чисел  $1^2 + 3^2 + \dots + K^2$  окажется больше А. Для решения этой задачи ученик написал программу, но, к сожалению, его программа – неправильная.

Pascal	Python
<pre>var a, s, k: integer; begin   read(A);   s := 0;   k := 1;   while s &lt;= A do begin     s := s + k*k;     k := k + 1;   end;   writeln(k) end.</pre>	<pre>A = int(input()) s = 0 k = 1 while s &lt;= A:     s = s + k*k     k = k + 1 print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 11.
  2. Укажите наименьшее значение А, при котором программа выведет верный ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 74) Дано целое положительное число А. Требуется вывести такое минимально возможное натуральное число К, при котором
- $$1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+K)$$
- окажется больше А. Для решения этой задачи ученик написал программу, но, к сожалению, его программа – неправильная.

Pascal	Python
<pre>var A, s, p, k: integer; begin   read(A);   s := 0; p := 0; k := 1;   while s &lt; A do begin     p := p + k;     s := s + p;     k := k + 1;   end;   writeln(k); end.</pre>	<pre>A = int(input()) s = 0 p = 0 k = 1 while s &lt; A:     p = p + k     s = s + p     k = k + 1 print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 12.
  2. Укажите любое значение А, при котором программа выведет верный ответ. Запишите этот ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 75) Дано целое положительное число А. Требуется вывести такое максимально возможное натуральное число К, при котором

$$1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+K)$$

не превышает А. Для решения этой задачи ученик написал программу, но, к сожалению, его программа – неправильная.

Pascal	Python
<pre>var A, s, p, k: integer; begin   read(A);   s := 0; p := 0; k := 1;   while s &lt;= A do begin     k := k + 1;     p := p + k;     s := s + p;   end;   writeln(k); end.</pre>	<pre>A = int(input()) s = 0 p = 0 k = 1 while s &lt;= A:   k = k + 1   p = p + k   s = s + p print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 12.
  2. Укажите все значения А, при которых программа выведет ответ 4.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 76) Даны натуральные числа А и В. Требуется найти такое минимально возможное натуральное число  $K \geq A$ , что сумма всех чисел от А до К больше или равна В. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre>var A, B, s, k: integer; begin   read(A, B);   s := 1;   k := A;   while s &lt;= B do begin     k := k + 1;     s := s + k;   end;   writeln(k); end.</pre>	<pre>A, B = map(int, input().split()) s = 1 k = A while s &lt;= B:   k = k + 1   s = s + k print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 5 и 15.
  2. Укажите любую пару значений А и В, при вводе которых программа выведет правильный ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 77) Даны натуральные числа А и В. Требуется найти такое минимально возможное натуральное число  $K \geq A$ , что сумма всех чисел от А до К больше, чем В. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre>var A, B, s, k: integer; begin   read(A, B);   s := 0;   k := A;   while s &lt; B do begin</pre>	<pre>A, B = map(int, input().split()) s = 0 k = A while s &lt; B:   k = k + 1   s = s + k</pre>

<pre> k := k + 1; s := s + k; end; writeln(k); end.</pre>	<pre> print(k)</pre>
---	----------------------

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 13 и 18.
  2. Укажите любую пару значений A и B, при вводе которых программа выведет правильный ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 78) Дано натуральное число A, большее 9, в записи которого нет нулей. Требуется найти максимальное двузначное число, которое можно увидеть в записи числа A. Например, для числа 18943 ответом будет число 94. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre> var A, n2: integer; begin   read(A);   n2 := 0;   while A &gt; 100 do begin     if A mod 100 &gt; n2 then       n2 := A mod 100;     A := A div 100;   end;   writeln(n2); end.</pre>	<pre> A = int(input()) n2 = 0 while A &gt; 100:     if A % 100 &gt; n2:         n2 = A % 100     A = A // 100 print(n2)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 9532.
  2. Укажите одно значение A, при вводе которого программа выведет правильный ответ. Укажите это ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 79) Дано натуральное число A, в записи которого нет нулей. Требуется найти минимальное двузначное число, которое можно увидеть в записи числа A. Например, для числа 815943 ответом будет число 15. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre> var A, n2: integer; begin   read(A);   n2 := 10;   while A &gt; 100 do begin     if A mod 100 &lt; n2 then       n2 := A mod 100;     A := A div 100;   end;   writeln(n2); end.</pre>	<pre> A = int(input()) n2 = 10 while A &gt; 100:     if A % 100 &lt; n2:         n2 = A % 100     A = A // 100 print(n2)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1245.

2. Укажите одно значение А, при вводе которого программа выведет правильный ответ. Укажите это ответ.

3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

80) На вход программы поступает натуральное число. Требуется определить сумму цифр в десятичной записи этого числа, которые не кратны 3 (считается, что 0 кратно любому числу). Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre> var N, digit, sum: longint; begin   readln(N);   sum := N mod 10;   while N &gt; 0 do begin     digit := N mod 10;     if digit mod 3 &gt; 0 then       sum := digit;     N := N div 10;   end;   if sum &gt; 0 then     writeln(sum)   else     writeln('NO') end. </pre>	<pre> N = int(input()) sum = N % 10 while N &gt; 0:     digit = N % 10     if digit % 3 &gt; 0:         sum = digit     N = N // 10 if sum &gt; 0:     print(sum) else:     print('NO') </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 654.

2. Укажите одно трёхзначное число, при вводе которого программа выведет правильный ответ. Укажите это ответ.

3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

81) На вход программы поступает натуральное число. Требуется определить сумму цифр в десятичной записи этого числа, которые кратны 3 (считается, что 0 кратно любому числу). Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre> var N, digit, sum: longint; begin   readln(N);   sum := N mod 10;   while N &gt; 0 do begin     digit := N mod 10;     if digit mod 3 = 0 then       sum := digit;     N := N div 10;   end;   if sum &gt; 0 then     writeln(sum)   else     writeln('NO') end. </pre>	<pre> N = int(input()) sum = N % 10 while N &gt; 0:     digit = N % 10     if digit % 3 == 0:         sum = digit     N = N // 10 if sum &gt; 0:     print(sum) else:     print('NO') </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 653.



2. Укажите одно трёхзначное число, при вводе которого программа выведет правильный ответ. Укажите это ответ.

3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

- 82) (Д.В. Богданов) Числа Фибоначчи задаются рекуррентным соотношением  $f(0) = 0, f(1) = 1$  и  $f(n) = f(n-1) + f(n-2)$  для всех целых  $n \geq 0$ . Требовалось написать программу, которая по входному целому неотрицательному значению  $n$ , не превосходящему 20, определяет соответствующее число Фибоначчи  $f(n)$ . Приведённая ниже программа содержит ошибки.

Pascal	Python
<pre>var n, i, f, fn : integer; begin   read(n);   fn := 1;   f := 1;   for i := 1 to n do begin     f := f + fn;     fn := f - fn;   end;   write(fn - 5) end.</pre>	<pre>n = int(input()) fn = 1 f = 1 for i in range(1, n+1):   f = f + fn   fn = f - fn print(fn - 5)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 4.

2. Укажите одно значение входной переменной  $n$ , при вводе которого программа выведет правильный ответ. Укажите это ответ.

3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

- 83) Дано натуральное число  $A > 0$ . Требуется вывести такое минимально возможное нечётное натуральное число  $K$ , при котором сумма  $1*2 + 3*4 + \dots + K*(K+1)$  окажется больше  $A$ . Приведённая ниже программа содержит ошибки.

Pascal	Python
<pre>var A, s, k: integer; begin   read(A);   s := 0;   k := 1;   while s &lt;= A do begin     k := k + 1;     s := s + k*(k+1);   end;   writeln(k) end.</pre>	<pre>A = int(input()) s = 0 k = 1 while s &lt;= A:   k = k + 1   s = s + k*(k+1) print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 15.

2. Укажите два наименьших значения  $A$ , при которых программа выведет верный ответ.

3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

- 84) Дано натуральное число  $A > 0$ . Требуется вывести такое минимально возможное нечётное натуральное число  $K$ , при котором сумма  $1*2 + 3*4 + \dots + K*(K+1)$  окажется больше  $A$ . Приведённая ниже программа содержит ошибки.

Pascal	Python
--------	--------

<pre> var A, s, k: integer; begin   read(A);   s := 0;   k := 0;   while s &lt;= A do begin     s := s + k*(k+1);     k := k + 1;   end;   writeln(k) end. </pre>	<pre> A = int(input()) s = 0 k = 0 while s &lt;= A:     s = s + k*(k+1)     k = k + 1 print(k) </pre>
---	---

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 15.
  2. Укажите два наименьших значения  $A$ , большие, чем 10, при которых программа выведет верный ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 85) (Д.В. Богданов) Требовалось написать программу, которая для введённого натурального числа  $n$  выводит общее количество его различных делителей, включая 1 и  $n$ . К сожалению, ученик написал программу с ошибками:

Pascal	Python
<pre> var n, j, k : integer; begin   read(n);   j := 2;   k := 0;   while j * j &lt; n do begin     if n mod j = 0 then       k := k + 2;     j := j + 1   end;   if j * j = n then     k := k + 3;   write(k) end. </pre>	<pre> n = int(input()) j = 2 k = 0 while j*j &lt; n:     if n % j == 0:         k = k + 2     j = j + 1 if j*j == n:     k = k + 3 print(k) </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 10.
  2. Укажите два наименьших значения входной переменной  $n$ , при вводе которого программа выведет правильный ответ. Укажите это ответ в каждом случае.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 86) На вход программы поступает натуральное число, не превышающее  $10^9$ . Нужно написать программу, которая выводит на экран максимальную цифру числа, меньшую 5. Если в числе нет цифр, меньших 5, требуется на экран вывести «NO». В приведённой программе есть ошибки.

Pascal	Python
<pre> var N, d, m: longint; begin   readln(N);   m := N mod 10;   while N &gt; 0 do begin </pre>	<pre> N = int(input()) m = N % 10 while N &gt; 0:     d = N % 10     if d &lt; 5: </pre>

<pre> d := N mod 10; if d &lt; 5 then     if d &gt; m then m := d; N := N div 10; end; if m = 0 then     writeln('NO') else writeln( m ) end. </pre>	<pre> if d &gt; m:     m = d N = N // 10 if m == 0:     print('NO') else:     print(m) </pre>
--	---

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 157.
  2. Укажите наибольшее трёхзначное значение входной переменной  $n$ , при вводе которого программа выведет правильный ответ. Укажите это ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 87) На вход программы поступает натуральное число, не превышающее  $10^9$ . Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 3. Если в числе нет цифр, кратных 3, требуется на экран вывести «NO». В приведённой программе есть ошибки.

Pascal	Python
<pre> var N, d, m: longint; begin     readln(N);     m := 0;     while N &gt; 0 do begin         d := N mod 10;         if d mod 3 = 0 then             if d &gt; m then m := d;         N := N div 10;     end;     if m = 0 then         writeln('NO')     else writeln( m ) end. </pre>	<pre> N = int(input()) m = 0 while N &gt; 0:     d = N % 10     if d % 3 == 0:         if d &gt; m:             m = d     N = N // 10 if m == 0:     print('NO') else:     print(m) </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 170.
  2. Укажите наименьшее трёхзначное значение входной переменной  $n$ , при вводе которого программа выведет правильный ответ. Укажите это ответ.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 88) (Досрочный ЕГЭ-2018) На вход программы поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество делящихся нацело на 4 чисел в исходной последовательности и максимальное делящееся нацело на 4 число. Если делящихся нацело на 4 чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно.

Pascal	Python
<pre> const n = 4; var i, x: integer; var maximum, count: integer; </pre>	<pre> n = 4 count = 0 maximum = 1000 </pre>

<pre> begin   count := 0;   maximum := 1000;   for i := 1 to n do begin     read(x);     if x mod 4 = 0 then begin       count := count + 1;       if x &lt; maximum then         maximum := x       end     end;   end;   if count &gt; 0 then begin     writeln(count);     writeln(maximum)   end   else     writeln('NO')   end. </pre>	<pre> for i in range (1, n+1):   x = int(input())   if x % 4 == 0:     count += 1     if x &lt; maximum:       maximum = x   if count &gt; 0:     print (count)     print (maximum) else:   print ("NO") </pre>
---	---

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности: 2 8 4 3.

2. Приведите пример такой последовательности, содержащей хотя бы одно делящееся нацело на 4 число, что при её вводе приведённая программа, несмотря на ошибки, выведет правильный ответ.

3. Найдите допущенные программистом ошибки и исправьте их.

Известно, что в тексте программы можно исправить ровно две строки так, чтобы она стала работать правильно.

- 89) Факториалом натурального числа  $n$  называется произведение всех натуральных чисел от 1 до  $n$ . Например,  $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ . На вход программы поступает положительное число  $A$ . Необходимо вывести минимальное натуральное  $K$ , для которого  $1! + 2! + \dots + K! > A$ . К сожалению, приведённая ниже программа неправильная.

Pascal	Python
<pre> var A, k, f, s: integer; begin   read(a);   k := 1;   f := 1;   s := 1;   while s &lt;= A do begin     f := f * k;     k := k + 1;     s := s + f   end;   writeln(k) end. </pre>	<pre> A = int(input()) k = 1 f = 1 s = 1 while s &lt;= A:   f *= k   k += 1   s += f print(k) </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 8.

2. Назовите минимальное  $A$ , большее 5, при котором программа выведет верный ответ.

3. Найдите допущенные программистом ошибки и исправьте их.

- 90) Факториалом натурального числа  $n$  называется произведение всех натуральных чисел от 1 до  $n$ . Например,  $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ . На вход программы поступает положительное число  $A$ .

Необходимо вывести минимальное натуральное  $K$ , для которого  $1! + 2! + \dots + K! > A$ . К сожалению, приведённая ниже программа неправильная.

Pascal	Python
<pre>var A, k, f, s: integer; begin   read(a);   k := 1;   f := 1;   s := 0;   while f &lt;= A do begin     k := k + 1;     f := f * k;     s := s + f;   end;   writeln(k) end.</pre>	<pre>A = int(input()) k = 1 f = 1 s = 0 while f &lt;= A:   k += 1   f *= k   s += f print(k)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 10.
  2. Назовите минимальное  $A$ , большее 15, при котором программа выведет неверный ответ.
  3. Найдите допущенные программистом ошибки и исправьте их.
- 91) На вход программы поступает натуральное число  $N$ , не превышающее  $10^9$ . Требуется найти и вывести наибольшую нечётную цифру в десятичной записи этого числа или вывести «NO», если таких цифр нет. К сожалению, приведённая ниже программа неправильная.

Pascal	Python
<pre>var N, d, m, t: longint; begin   read(N);   t := 0;   m := t;   while N &gt; 1 do begin     d := N mod 10;     if (d mod 2 = 1) or (d &gt; m) then       m := d;     N := N div 10;   end;   if m = t then     writeln('NO')   else writeln(m) end.</pre>	<pre>N = int(input()) t = 0 m = t while N &gt; 1:   d = N % 10   if d % 2 == 1 or d &gt; m:     m = d   N = N // 10 if m == t:   print("NO") else:   print(m)</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 815.
  2. Какое наименьшее число может быть выведено при работе этой программы? Приведите пример числа  $N$ , при вводе которого программа выведет такой ответ.
  3. Найдите допущенные программистом ошибки и исправьте их.
- 92) На вход программы поступает натуральное число  $N$ , не превышающее  $10^9$ . Требуется найти и вывести наибольшую чётную цифру в десятичной записи этого числа или вывести «NO», если таких цифр нет. К сожалению, приведённая ниже программа неправильная.

Pascal	Python
<pre>var N, d, m, t: longint; begin</pre>	<pre>N = int(input()) t = 0</pre>

<pre> read(N) ; t := 0; m := t; while N &gt; 1 do begin   d := N mod 10;   if (d mod 2 = 0) or (d &gt; m) then     m := m + d;   N := N div 10 end; if m = t then   writeln('NO') else writeln(m) end. </pre>	<pre> m = t while N &gt; 1:   d = N % 10   if d % 2 == 0 or d &gt; m:     m = m + d   N = N // 10 if m == t:   print("NO") else:   print(m) </pre>
---	--

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 132.
  2. Какое наибольшее число может быть выведено при вводе трёхзначного числа? Сколько существует таких чисел N, при вводе которого программа выведет такой ответ?
  3. Найдите допущенные программистом ошибки и исправьте их.
- 93) (В. Бабий) Дано целое положительное число N, не превосходящее 1000. Необходимо определить, является ли это число степенью числа 5. То есть требуется определить, существует ли такое целое число T, что  $5^T = N$ , и вывести это число либо сообщение, что такого числа не существует. Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной.

Pascal	Python
<pre> var   n, s, t: integer; begin   readln(n);   s := 0;   t := -1;   while n &gt; 0 do   begin     s := s + n mod 5;     t := t + 1;     n := n div 5;   end;   if t = 1 then     writeln(s)   else     writeln('Не существует'); end. </pre>	<pre> n = int(input()) s = 0; t = -1; while n &gt; 0:   s += n % 5   t += 1   n = n // 5 if t == 1:   print(s) else:   print('Не существует') </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 125.
  2. Приведите пример числа N, которое является степенью числа 5, и при вводе которого программа будет выводить правильный результат.
  3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.
- 94) (А. Богданов) Заказчику потребовалась программа, которая анализирует десятичные числа. Если все цифры числа идут в строго возрастающем порядке, тогда программа должна вывести «1»,

иначе «0». Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Pascal	Python
<pre> var a,b,d:integer; begin   read(a);   b := 9;   while a &gt; 0 do begin     d := a mod 10;     if d &gt; b then break;     a := a div 10;     b := d;   end;   if a=0 then writeln(1)   else writeln(0); end. </pre>	<pre> a = int(input()) b = 9 while a &gt; 0:   d = a % 10   if d &gt; b: break   a = a // 10   b = d if a == 0:   print(1) else:   print(0) </pre>
C++	
<pre> #include &lt;iostream&gt; using namespace std; int main() {   int a, b, d;   cin &gt;&gt; a;   b = 9;   while (a &gt; 0) {     d = a % 10;     if (d &gt; b) break;     a = a / 10;     b = d;   }   if (a == 0) cout &lt;&lt; 1;   else cout &lt;&lt; 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1234.
2. Приведите пример максимального десятичного трехзначного числа а, для которого программа выведет верный ответ «1».
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.