

ПОДГОТОВКА К ЕГЭ
ПО ИНФОРМАТИКЕ И ИКТ
ЧАСТЬ С
(ЗАДАНИЕ С1)

Автор Воеводина Е.Т.
учитель информатики и ИКТ
ГОО СОШ № 603
Фрунзенского района
г. Санкт-Петербурга

Задание С1 – на поиск ошибок и оценивается в 3 балла

ВОЗМОЖНЫЕ ВАРИАНТЫ ОШИБОК:

- ✓ *в программе осуществлен ввод лишних данных;*
- ✓ *ошибка в операторе условия (> вместо <, + вместо -) или в условии выхода из области;*
- ✓ *несоответствие количества «скобок» в условии (while и wend, ошибка до или после else);*
- ✓ *возможно строки в программе поменяны местами (или строка удалена)*

Классификация заданий С1

- 1) *входные данные – числовые выражения и задание содержит три вопроса:*
 - *определить при каких входных данных программа работает неверно (вопрос для всех классов заданий С1);*
 - *определить лишнюю часть представленной программы;*
 - *исправить предложенную программу.*
- 2) *входные данные – числовые выражения, но вопросов два (т.е. в исходной части программы две ошибки);*
- 3) *входные данные – числовые, вопросов два, оба касаются исправления программы, но второй вопрос сформулирован довольно жёстко (требует не содержать логических функций и т.д.)*
- 4) *входные данные – строковые переменные, а в остальном соответствует первому классу заданий С1.*

Некоторые рекомендации для подготовки

Важно ориентировать учеников на то, что в этом задании отдельно оценивается ответ на каждый поставленный вопрос.

В первом вопросе необходимо указать, как правило, примеры входных данных, при которых программа будет работать неправильно!

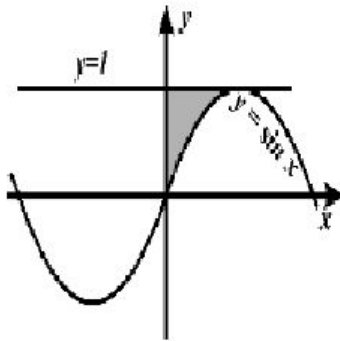
Ответ не должен состоять из короткого «да» или «нет». Если требуется обоснование ответа, оно должно быть приведено.

*Если задания перекликаются, например, во 2 задании предлагается усовершенствовать программу любым способом, а в 3 – конкретным (без использования *AND* и *OR*), то правильный ответ на более сложный вопрос может дать 2 балла и перекрыть таким образом оба задания.*

Эффективность программы в задании не оценивается.

Рассмотрим несколько примеров заданий С1 с различными формулировками условия

C1



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.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>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. end.</pre>	<pre>INPUT x, y IF y<=1 THEN IF x>=0 THEN IF y>=SIN(x) THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (y<=1) if (x>=0) if (y>=sin(x)) printf("принадлежит"); else printf("не принадлежит"); } }</pre>

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.)

Содержание верного ответа и указания по оцениванию
(допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) Пример: $x=3$, $y=0,5$ (Любая пара (x,y) , для которой выполняется: $y>1$ или $x<0$ или $(y>=\sin x$ и $x>\pi/2$ и $y<=1)$)

2) Возможная доработка (Паскаль):

```
if (y<=1) and (x>=0) and (y>=sin(x)) and (x<=3,14/2) then  
  write('принадлежит')  
else  
  write('не принадлежит')
```

(могут быть и другие способы доработки).

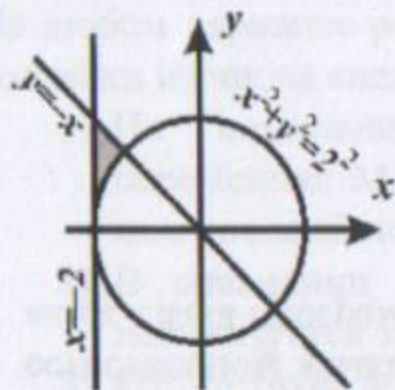
Обратите внимание! В задаче требовалось выполнить **три** действия: указать пример входных данных, при которых программа работает неверно, и исправить две ошибки:

1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).

2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых $(y>=\sin x$ и $x>=\pi/2$ и $y<=1)$.

Т.е. в программе не учтено условие $x<=\pi/2$

С1



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.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>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. </pre>	<pre>INPUT x, y IF x*x+y*y>=4 THEN IF x>= -2 THEN IF y<= -x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END </pre>	<pre>void main(void) { float x,y; scanf("%f%f",&x,&y); if (x*x+y*y>=4) if (x>= -2) if (y<= -x) printf("принадлежит"); else printf("не принадлежит"); } </pre>

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).

Элементы ответа:

1) Пример: $x = -1$, $y = -3$ (Любая пара (x, y) , для которой выполняется: $x^2 + y^2 < 4$ или $x < -2$ или $(y < 0$ и $y \leq -x)$)

2) Возможная доработка (Паскаль):

```
if (x*x+y*y>=4) and (x>= -2) and (y<= -x) and (y>=0) then  
  write('принадлежит')  
else  
  write('не принадлежит')
```

(могут быть и другие способы доработки).

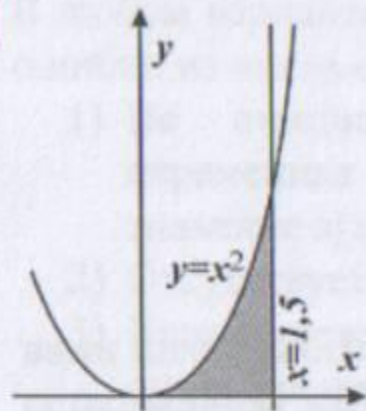
Обратите внимание! В задаче требовалось выполнить **три** действия: указать пример входных данных, при которых программа работает неверно и исправить две ошибки:

1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).

2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых $(x^2 + y^2 >= 4)$ и $(y \leq -x)$ и $(x >= -2)$ и $(y < 0)$.

Т.е. в программе не учтено условие $y >= 0$

C1



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.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if y<=x*x then if x<=1.5 then if y>=0 then write('принадлежит') else write('не принадлежит'); end.</pre>	<pre>INPUT x, y IF y<=x*x THEN IF x<=1.5 THEN IF y>=0 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f%f", &x, &y); if (y<=x*x) if (x<=1.5) if (y>=0) printf("принадлежит"); else printf("не принадлежит"); }</pre>

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).

Элементы ответа:

1) Пример: $x = -2, y = 1$

(Любая пара (x, y) , для которой выполняется: $x > 1.5$ или $(y \geq 0$ и $x < 0$ и $y \leq x^2)$ или $y > x^2$)

2) Возможная доработка (Паскаль):

```
if (y <= x*x) and (x <= 1.5) and (y >= 0) and (x >= 0) then  
  write('принадлежит')  
else  
  write('не принадлежит');
```

(могут быть и другие способы доработки).

Обратите внимание! В задаче требовалось выполнить три действия: указать пример входных данных, при которых программа работает неверно, и исправить две ошибки:

1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).

2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых $(y \leq x^2)$ и $(y \geq 0)$ и $(x < 0)$.

Т.е. в программе не учтено условие $x \geq 0$

С1. 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).

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)

Программа на языке Паскаль	Программа на языке Бейсик
<pre>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.</pre>	<pre>PRINT "Искомые поля" FOR J=5 TO 8 FOR I=1 TO 8 IF (I=9-J) OR (I=J) THEN PRINT "I="; I PRINT "J="; J ENDIF NEXT I NEXT J END</pre>

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) Поле $i=5, j=4$ будет выдано приведенной программой

(т.к. $i+j=9$) (необходимое пояснение!)

2) $(i=1, j=8), (i=2, j=8), (i=5, j=5)$ – эти поля удовлетворяют постановке задачи (ответ очевиден, если нарисовать шахматную доску 8×8 и королей в двух нижних углах)

3) Возможный (самый короткий) способ доработки:

вместо `if (i=9-j) OR (i=j)`

проверять условие `if (i>=9-j) AND (i<=j)`

	1	2	3	4	5	6	7	8
8								
7								
6								
5								
4								
3								
2								
1								

C1. Требовалось написать программу, в которой нужно было проверить, лежит ли число x на числовой оси между числами a и b ("между" понимается в строгом смысле, т.е. случай $x=a$ или $x=b$ недопустим). Числа x , a , b являются натуральными, и известно, что a отлично от b (но неизвестно: $a>b$ или $b>a$). Входная информация вводится с клавиатуры, а на выходе должно быть сообщение вида « x между a и b » (если это действительно так), в противном случае никакой выходной информации не выдается.

Программист торопился и написал программу некорректно.

Последовательно выполните три задания:

- 1) Приведите пример таких чисел a , b , x , при которых программа работает неправильно.**
- 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, соблюдая дополнительное условие: доработанная программа**

не должна использовать логических операций AND или OR.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ
VAR a,b,x: integer;	CLS
 p: integer;	INPUT a, b, x
BEGIN	IF (a>x) AND (x>b) THEN
 readln(a,b,x);	 PRINT "x между a, b"
 if (a>x) AND (x>b) then	END
 writeln('x между a,b');	
END.	

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Элементы ответа:

1) Пример: $a=1$ $x=2$ $b=3$

2) Возможная доработка:

```
if a<b then begin p:=a; a:=b; b:=p end;
```

```
if (a>x) AND (x>b) then
```

```
writeln(' x между a,b');
```

(могут быть и другие правильные способы доработки).

3) Возможная доработка без использования логических операций AND, OR:

```
p:=(x-a)*(x-b); if p<0 then
```

```
writeln(' x между a,b');
```

(могут быть и другие способы доработки с соблюдением дополнительного условия).

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

Программа на Паскале	Программа на Бейсике	Программа на Си
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then write('любое число') else if a > 0 then write('x >', -b/a) else write('x <', -b/a); end.</pre>	<pre>INPUT a, b, x IF a = 0 THEN PRINT "любое число" ELSE IF a > 0 THEN PRINT "x > ", -b/a ELSE PRINT "x < ", -b/a END</pre>	<pre>void main(void) { float a,b,x; scanf("%f%f%f",&a,&b,&x); if (a=0) printf(«любое число»); else if (a > 0) printf("x > %f", -b/a); else printf("x < %f", -b/a); }</pre>

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.

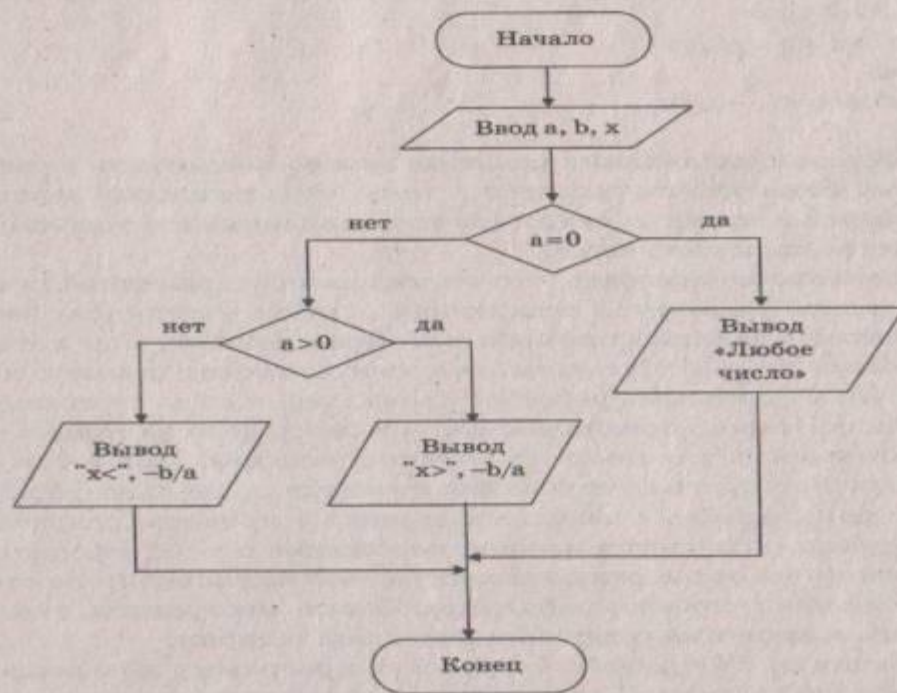
3) Укажите, как надо доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

Прежде, чем приступить к решению задачи, следует ещё раз обратить внимание на распространённую ошибку учащихся. Вместо требуемого в задании примера исходных данных, при которых программа работает неправильно, они приводят пример, когда ошибочная программа работает правильно.

Возможно это связано с тем, что учащимися не всегда правильно понимается, что роль тестовых данных в программировании, не только доказать работоспособность программы, но и выявить содержащиеся в ней ошибки.

Решение

Рассмотрим текст программы на языке Паскаль. Составим блок-схему этой программы.



По созданной блок-схеме можно сделать следующие выводы о логических ошибках в программе.

Ввод значения x является излишним, так как впоследствии x либо заново вычисляется, либо не используется, поскольку относится к выходным, а не входным данным.

В случае $a = 0$ не анализируется знак переменной b , поэтому при любом неположительном значении b программа будет давать неверный ответ.

Один из вариантов доработки программы:

```
readln(a,b);
if a=0 then
  if b>0 then
    write('любое число')
  else
    write('нет решений')
else
  if a>0 then
    write('x>', -b/a)
  else
    write('x<', -b/a);
```

C1

Требовалось написать программу, которая считает во введенной с клавиатуры строке, содержащей только буквы и пробелы (' '), количество слов, начинающихся и заканчивающихся одной и той же буквой. Слово, состоящее из одной буквы, также считается. Программист написал программу неправильно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ
<pre> Var s:string; a:char; i, k: integer; begin readln(s); k:=0; for i:=1 to length(s) do begin if (s[i] = ' ') and (s[i+1]<>' ') then a:=s[i+1]; if (s[i] <>' ') and (s[i+1] = ' ') then if s[i] = a then k:=k+1; if (s[i-1] = ' ') and (s[i] <>' ') and (s[i+1] = ' ') then k:=k+1; end; if k = 0 then write ('таких слов нет') else write ('k = ', k) end. </pre>	<pre> Input st\$ K = 0 FOR i = 1 to LEN(st\$) IF (MID\$(st\$,i,1) = " ") AND (MID\$(st\$,i,1) <> " ") THEN a\$ = MID\$(st\$,i,1) IF (MID\$(st\$,i,1) <> " ") AND (MID\$(st\$,i,1) = " ") THEN IF MID\$(st\$,i,1) = a\$ THEN k=k+1 IF (MID\$(st\$,i-1,1) = " ") AND (MID\$(st\$,i,1) <> " ") AND (MID\$(st\$,i+1,1) = " ") THEN k=k+1 ENDIF NEXT i IF k=0 THEN PRINT "таких слов нет" ELSE PRINT k ENDIF END </pre>

Последовательно выполните три задания:

- 1) Приведите пример таких строк, 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).

Этот вариант задания оказался «сырым». Если проанализировать программы на Паскале и Бейсике, выяснится, что они разные. Оказывается, что в программе на Бейсике ошибок больше.

Для определённости рассмотрим программу на Паскале. Анализ программы показывает, что она даёт неправильный результат всегда, когда строка начинается не с пробела или не заканчивается пробелом.

1. Ответ на первый вопрос, например, последовательность: b aba

2. Лишняя часть программы:

```
if (s[i-1]=' ')and (s[i]<>' ')and(s[i+1]=' ')  
then k:=k+1;
```

*3. Доработать программу можно следующим образом:
перед циклом добавить :*

```
a:=s[1]; {сохранение первого символа}  
a во второе ветвление добавить ещё одно условие –  
if (s[i]<>' ')and(s[i+1]=' ')or(i=length(s)) then if s[i]=a
```

Таким образом вариант доработанной программы может выглядеть так:

```
Var s:string;  
a: char;  
l, k: integer;  
begin  
  readln(s); k:=0;  
  a:=s[1]; {сохранение значения первого символа}  
  for i:=1 to length(s) do  
    begin  
      if (s[i]=` `) and (s[i+1]<>` `) then a:=s[i+1];  
      if (s[i]<>` `) and (s[i+1]=` `) or (i=length(s)) then if s[i]=a  
        then k:=k+1;  
    end;  
  if k=0 then  
    write (`таких слов нет`)  
  else  
    write (`k=`, k)  
end.
```

ЛИТЕРАТУРА

1. Информатика и ИКТ. Подготовка к ЕГЭ под ред. Н.Макаровой
2. ЕГЭ. Универсальные материалы для подготовки учащихся. Учебник ФИПИ.
3. Ссылка на прохождение ЕГЭ по информатике:
<http://www.bitner.ru/demo-ege/>
4. <http://vmatykin.narod.ru/spravka/011.htm#mod> Справочник начинающего программиста
5. http://wiki.lik590.ru/doku.php/primernye_temy_kursovyx_rabot.2008-2009:poleznye_ssylnki (Учебник по VB, Основы алгоритмизации, решение задач, экзамены, олимпиады, сценарии уроков)
6. <http://www.metod-kopilka.ru/page-6-1-4/html> копилка педагогического опыта
7. Сайт Константина Полякова с подробным анализом задач по ЕГЭ:
<http://kpolyakov.narod.ru/school/ege.htm>