Решение задач С1 части С Единого государственного экзамена

Автор – Богачёва Г.В.
Учитель информатики
Лицей 144 Санкт-Петербурга

необходимо:

- Проанализировать геометрический чертёж, выделив условия, ограничивающие заштрихованную область (причём в задаче 2010 года, а также в демоверсии 2011 года условие задачи значительно усложнилось с математической точки зрения)
- Выяснить, как связаны между собой эти условия (если областей несколько, определить, в каких отношениях находятся фигуры – разбиение, объединение, исключение и т.д), другими словами, составить сложное условие, описывающее выделенную область.

необходимо:

- Проанализировать данную в условии программу, выявив ошибки: недостаток условий и ошибку в алгоритме, связанную с использованием вложенных условных операторов.
- Чётко ответить на вопросы задачи: указать координаты точки, для которой программа не выводит «не принадлежит». Предложить метод доработки, лучше с использованием сложных условий, это исключит ошибки при написании вложенных условных операторов.

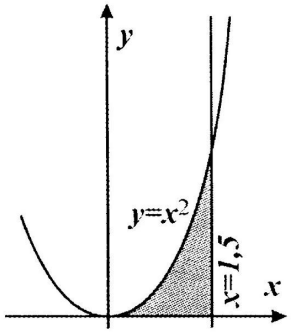
необходимо:

При решении задач этого типа основные методы –анализ и синтез.

При подготовке к решению задач последовательно изучаются следующие темы: правила построения программы на изучаемом языке, правила работы с переменными, полная и неполная формы ветвления, использование операторных скобок begin end в операторах ветвления, составление сложных условий с помощью логических операций и, наконец, использование вложенных условных операторов. Весь материал по этой теме блестяще изложен у К.Ю. Полякова, презентациями и сайтом которого я регулярно пользуюсь.

Условие задачи 1:

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

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.
	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
		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");
		}

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 задачи 1:

- не обращая внимание на программу, анализируем чертёж, выявив условия, которым отвечают координаты точек заштрихованной области.
- заштрихованная область находится

1. справа от оси абцисс, что равносильно **первому условию** $x \geq 0$ (с учетом границы здесь и далее получаем нестрогие неравенства)
2. слева от линии $x = 1{,}5$, поэтому получаем **второе условие** $x \leq 1{,}5$

Решение задачи 1:

- заштрихованная область ограничена с двух сторон по координате ординат: она находится

1. ниже линии y <= x^2, откуда следует **третье условие**

$$y \le x^2$$

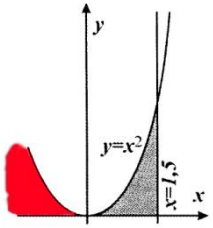
2. выше линии $y = 0$, что дает **четвертое условие**

$$y \ge 0$$

- значит, точка находится в заданной области, если все эти четыре условия выполняются одновременно; отсюда следует, что в программе нужно использовать четыре вложенных условных оператора или один условный оператор со сложным условием, в котором четыре простых условия связаны с помощью логической операции **and** («И», одновременное выполнение всех условий)

Решение задачи 1:

- анализируем программу: здесь только три вложенных условных оператора с простыми отношениями, поэтому какое-то условие не учтено; определяем, что не учтено условие $x \geq 0$
- оператор **write('принадлежит')** помещен внутрь всех трех условных операторов, то есть, он выполнится тогда, когда только три условия истинны;
- отметим на рисунке область, где выполняются все нужные условия, кроме $x \geq 0$ (красная зона);



Решение задачи 1:

- для всех точек, которые находятся в «красной» зоне программа выдаст сообщение «принадлежит», хотя в самом деле эти точки не принадлежит заданной области; одна из таких точек имеет координаты

  x = -2, y = 1

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

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

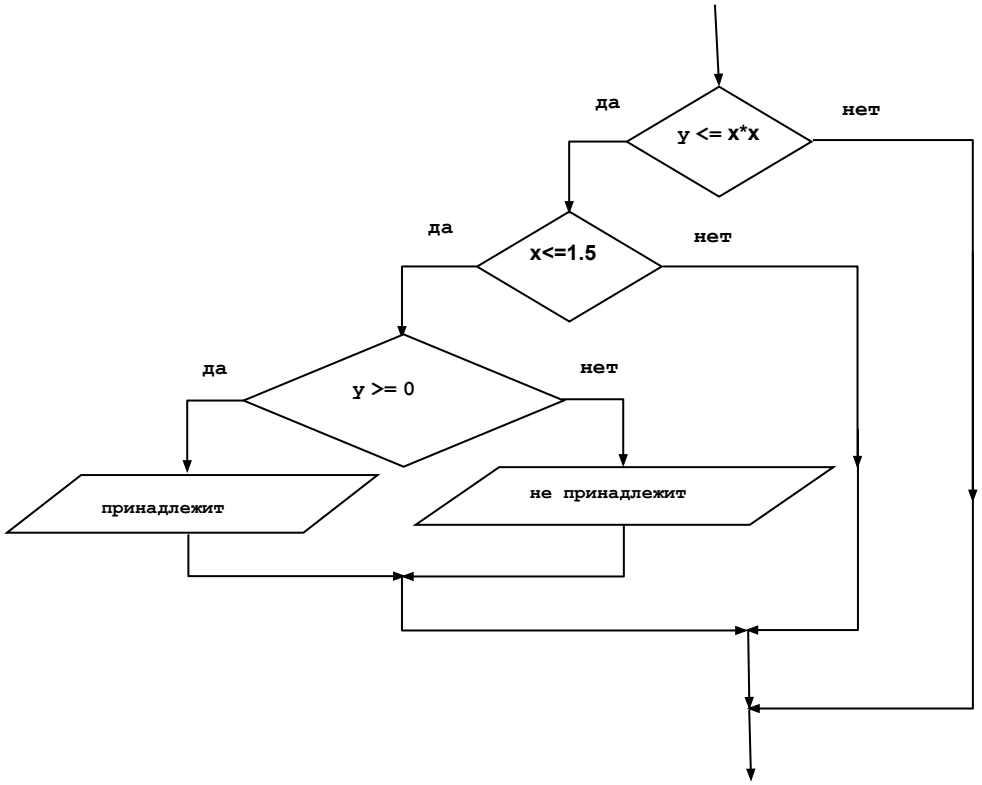
Решение задачи 1:

- можно подумать, **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 <= x*x then
  if x<=1.5 then
    if y >= 0 then
        write('принадлежит')
    else write('не принадлежит')
```

Решение задачи 1:

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



Решение задачи 1:

- по схеме видим, что при $y > x^2$ (первое условие ложно), а также при $x > 1.5$ (второе условие ложно) программа вообще не выдает никакого сообщения, то есть, работает неправильно; таким образом, координаты любой точки, для которой $y > x^2$ или $x > 1.5$, могут быть указаны в ответе как пример набора входных данных, при которых программа работает неправильно x=2, y=6
- итак, первая часть ответа

  примеры входных данных, на которых программа работает неверно:
  (x= -2, y= 1) (неправильно определяет принадлежность точки области)
  (x=2, y=6) или **(x= 5, y=0)** (не выдает вообще никакого сообщения)

Решение задачи 1:

- остается исправить эту программу; 1 способ: добавим в программу четвертый (вложенный) условный оператор, проверяющий условие, и еще три блока **else**, чтобы выводить строку «не принадлежит» в том случае, когда хотя бы один из них не сработал:

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

Решение задачи 1:

- 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
- хотя приведенный выше метод дает работоспособную программу, она нерациональна для такой задачи; видно, что оператор **write('не принадлежит')** повторяется в тексте 4 раза

Решение задачи 1:

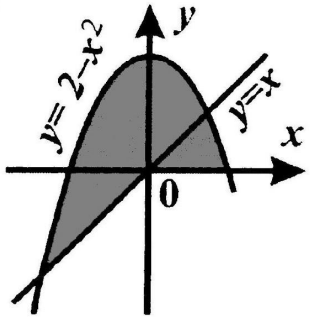
- более элегантное решение формулируется на словах так: «точка принадлежит области, если выполняются одновременно 4 приведенных выше условия, а иначе – не принадлежит»; а вот реализация на Паскале:

```
var x,y: real;
begin
readln(x,y);
if (x >= 0) and (y <= x*x)  and  (x<=1.5) and (y >= 0) 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)

Условие задачи 2:

C1



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

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ

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.
	INPUT x, y
	IF y>=x THEN
	IF y>=0 THEN
	IF y<=2-x*x THEN
	PRINT "принадлежит"
	ELSE
	PRINT "не принадлежит"
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{ float x,y;
		scanf("%f%f",&x,&y);
		if (y>=x)
		if (y>=0)
		if (y<=2-x*x)
		printf("принадлежит");
		else
		printf("не принадлежит");
		}

Последовательно выполните следующее:

1) Приведите пример таких чисел x, y, при которых программа неправильно 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 правильный способ доработки исходной программы).

Решение задачи 2:

Изложим решение менее подробно, чем в первой задаче.

- Анализируем чертёж, определяем условия: разбиваем заштрихованную область на 2 части прямой x=0:

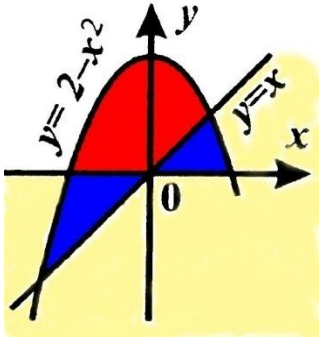
  1 фигура: y <= 2-x^2, $y \geq 0$ $x \geq 0$. Точка находится в заданной области, если все эти три условия выполняются одновременно.

  2 фигура: $x \leq 0$ $y \geq x$, y <= 2-x^2 . Точка находится в заданной области, если все эти три условия выполняются одновременно.

- Объединяем две пересекающиеся области с помощью логической операции дизъюнкции.

Решение задачи 2:

- Анализируем приведённую в задании программу. Отметим на рисунке область, где выполняются все условия, проверка которых есть в программе, красным цветом. Видно, что точки частей выделенной области, закрашенных синим цветом, в проверяемую область не входят, и, следовательно, программа ошибочно даст «не принадлежит». Координаты одной из таких точек $x = 0{,}5$, $y = 0{,}2$



Решение задачи 2:

- Теперь выясним, когда программа выдает сообщение «не принадлежит». Анализируя вложенные условия, видим, что ветвь else относится к внутреннему оператору, соответственно, при y <= x (первое условие ложно), а также при y <= 0 (второе условие ложно) программа вообще не выдает никакого сообщения, то есть, работает неправильно; таким образом, координаты любой точки, для которой y <= x или

  y <= 0 , могут быть указаны в ответе как пример набора входных данных, при которых программа работает неправильно, например, x=1, y= -3

Решение задачи 2:

- Первая часть ответа такова: примеры входных данных, на которых программа работает неверно:
  (x = 0,5, y = 0,2**)** (неправильно определяет принадлежность точки области)
  (x=1, y= -3**)** (не выдает вообще никакого сообщения)
- Приведём наиболее рациональную форму доработки программы, использование которой исключает ошибки при написании вложенных условных операторов:

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

Источники:

1. http://kpolyakov/narod.ru (сайт К.Ю. Полякова)
2. Условия задач взяты из экзаменационных материалов прошлых лет.