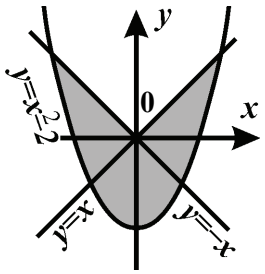
C1



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

Ответ:

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

Элементы ответа:

1) Пример: x=2, y=2 (Любая пара (x,y), для которой выполняется: y>x или y>–x)

2) Возможная доработка (Паскаль, разбиение области на две части прямой x=0):

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

Возможная доработка (Си, разбиение на две пересекающиеся области):

```
if (y>=x*x-2 && (y<=x || y<=-x))
   printf("принадлежит");
else
   printf("не принадлежит");
```

Возможная доработка (Бейсик, отбрасывание части от большей области, используются вложенные условия):

```
IF y >= x * x - 2 THEN
  IF NOT (y > x AND y > -x) THEN
    PRINT "принадлежит"
  ELSE
    PRINT "не принадлежит"
  ENDIF
ELSE
  PRINT "не принадлежит"
ENDIF
```

Обратите внимание, что вариантов доработки может быть достаточно много, но обычно правильное описание заштрихованных областей в них представляет собой

- или объединение двух (или более) возможно пересекающихся областей,
- или исключение одной области из другой.

При разделении области вдоль какой-либо линии точки, которые лежат на этой линии внутри области, могут быть причислены к одной части, к другой или к обеим (то есть, например, в приведенном решении на языке Паскаль одно из условий `x>=0` или `x<=0` может быть строгим).

Могут быть и другие верные способы доработки.

Указания по оцениванию	**Баллы**
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, второго или третьего условия программа не выдавала ничего (отсутствуют случаи ELSE). 2. Приведенным трем ограничениям не удовлетворяют точки плоскости, у которых ($y>-x$) и ($y\geq x^2-2$) и ($y\leq x$), а также точки, у которых ($y>x$) и ($y\geq x^2-2$) и ($y\leq -x$).	
Правильно выполнены все **три** действия. Исправлены обе ошибки. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	3
1. Правильно выполнены два действия из трех (исправлены обе ошибки, но не указан/неправильно указан пример требуемых входных данных, либо правильно указан пример входных данных, программа правильно работает при большем числе случаев, чем исходная, но не при всех). Например, выдает "не принадлежит" и для всех точек, у которых ($y>x$) и ($y>-x$). При этом не допускается, чтобы программа неправильно работала при тех входных данных, при которых раньше работала правильно. ИСКЛЮЧЕНИЕ! При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо "`y<=x`" используется "`y<x`" (даже если программа при этом стала неверно работать при тех входных данных, при которых раньше работала правильно). 2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражениях).	2
Правильно выполнено только одно действие из трех. То есть, либо только приведен пример входных данных, либо он не приведен (или приведен неверно), но имеется программа, корректно работающая при большем количестве входных данных, чем исходная, но не при всех. При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства, или наоборот).	1
Все пункты задания выполнены неверно (пример входных	0

данных не указан или указан неверно, программа не приведена, либо приведенная программа корректно работает в не большем количестве случаев, чем исходная).	
Максимальный балл	*3*

C2 Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое элементов массива, имеющих нечетное значение. Гарантируется, что в исходном массиве хотя бы один элемент имеет нечетное значение.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль

```
const
    N=30;
var
    a: array [1..N] of integer;
    i, x, y: integer;
    s: real;
begin
   for i:=1 to N do readln(a[i]);
   …

end.
```

Бейсик

```
N=30
DIM A(N) AS INTEGER
DIM I, X, Y AS INTEGER
DIM S AS SINGLE
FOR I = 1 TO N
INPUT A(I)
NEXT I

…

END
```

Си

```
#include <stdio.h>
#define N 30
void main(void)
{int a[N];
int i, x, y;
float s;
for (i=0; i<N; i++)
    scanf("%d", &a[i]);
…

}
```

Естественный язык

Объявляем массив A из 30 элементов.
Объявляем целочисленные переменные I, X, Y.
Объявляем вещественную переменную S.
В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й.

…

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учетом синтаксиса и особенностей используемого вами языка программирования.

Ответ:

Содержание верного ответа и указания по оцениванию **(допускаются иные формулировки ответа, не искажающие его смысла)**	
На языке Паскаль	**На языке Бейсик**

`x:=0;`
`y:=0;`
`for i:=1 to N do`
`if (a[i] mod 2=1) then begin`
`  x:=x+a[i];`
`  y:=y+1;`
`end;`
`s:=x/y;`
`writeln(s);`
	`X = 0`
	`Y = 0`
	`FOR I = 1 TO N`
	`IF A(I) MOD 2 = 1 THEN`
	`X = X + A(I)`
	`Y = Y + 1`
	`ENDIF`
	`NEXT I`
	`S = X / Y`
	`PRINT S`

На языке Си	**На естественном языке**

`x=0;`
`y=0;`
`for (i=0; i<N; i++)`
`  if (a[i]%2==1)`
`  { x=x+a[i];`
`  y++;`
`  }`
`s=(float)x/y;`
`printf("%f", s);`
	Записываем в переменные X и Y начальное значение, равное нулю. В цикле от первого элемента до тридцатого находим остаток от деления элемента исходного массива на два. Если этот остаток равен единице, то увеличиваем счетчик суммы X на значение текущего элемента массива, а счетчик количества Y на 1. Переходим к следующему элементу. После цикла производим деление счетчика суммы X на счетчик количества Y и записываем результат в переменную S. Выводим значение переменной S.

Указания по оцениванию	**Баллы**
Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.	2

В любом варианте решения может присутствовать не более одной ошибки из числа следующих: 1) Значения переменных X и Y находятся верно, однако среднее арифметическое считается неверно (например, производится действие X/N или неверно происходит преобразование типов при делении). 2) Неверно осуществляется проверка значения элемента массива на нечетность. 3) Не инициализируются или неверно инициализируются переменные X и Y. 4) Отсутствует вывод ответа. 5) Используется переменная, не объявленная в разделе описания переменных. 6) Не указано или неверно указано условие завершения цикла. 7) Индексная переменная в цикле не меняется (например, в цикле while). 8) Неверно расставлены операторные скобки.	1
Ошибок, перечисленных в п. 1-8, две или больше, или алгоритм сформулирован неверно.	0
Максимальный балл	*2*

С3 Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 3, а во второй 4 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то куче или добавляет 4 камня в какую-то кучу. Игрок, после хода которого общее число камней в двух кучах становится больше 25, **проигрывает**. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Ответ:

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

Выигрывает второй игрок.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры, в первой и второй кучах соответственно.

	1 ход	2 ход	3 ход	4 ход	
Стартовая позиция	**I-й игрок (все варианты хода)**	**II-й игрок (выигрышный ход)**	**I-й игрок (все варианты хода кроме непосредственно проигрышных)**	**II-й игрок (выигрышные ходы, экзаменуемому достаточно указать один из вариантов)**	**Пояснение**
3, 4	6, 4	**12, 4**	12, 8	**16, 8** **12, 12**	Любой следующий ход первого игрока является непосредственно проигрышным
			16, 4	**20, 4** **16, 8**	
	3, 8	**3, 12**	6, 12	**10, 12** **6, 16** **12, 12**	
			3, 16	**7, 16** **3, 20** **6, 16**	
			7, 12	**11, 12** **7, 16**	
	7, 4	**11, 4**	15, 4	**19, 4** **15, 8**	
			11, 8	**15, 8** **11, 12**	

Таблица содержит ***все возможные*** варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

Указания по оцениванию	**Баллы**
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры).	3
При наличии в представленном решении одного из пунктов: 1. Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности. 2. Правильно указан выигрывающий игрок, приведено дерево игры, но отсутствует обоснование правильности выигрывающей стратегии. 3. Правильно указан выигрыш второго игрока, рассмотрены все варианты хода первого игрока, для каждого из них правильно указан выигрывающий ответ второго игрока. Однако анализ игры не доведен до конца и отсутствует обоснование стратегии.	2
При наличии в представленном решении одного из пунктов: 1. Правильно указаны все варианты хода первого игрока и возможные ответы второго игрока (в том числе и все выигрышные), но неверно определены дальнейшие действия и неправильно указан победитель. 2. Правильно указан выигрыш второго игрока, но описание выигрышной стратегии неполно и для некоторых (больше одного, но не всех) вариантов хода первого игрока правильно указан выигрывающий ответ второго игрока.	1
Задание не выполнено, или в представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока).	0
Максимальный балл	*3*

C4 На вход программе подается набор символов, заканчивающийся точкой (в программе на языке Бейсик символы можно вводить по одному в строке, пока не будет введена точка, или считывать данные из файла). Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая сначала будет определять, есть ли в этом наборе символы, соответствующие десятичным цифрам. Если такие символы есть, то можно ли переставить их так, чтобы полученное число было симметричным (читалось одинаково как слева направо, так и справа налево). Ведущих нулей в числе быть не должно, исключение – число 0, запись которого содержит ровно один ноль.

Если требуемое число составить невозможно, то программа должна вывести на экран слово “NO”. А если возможно, то в первой строке следует вывести слово “YES”, а во второй – искомое симметричное число. Если таких чисел несколько, то программа должна выводить максимальное из них. Например, пусть на вход подаются следующие символы:

Do not 911 to 09 do.

В данном случае программа должна вывести

YES

91019

Ответ:

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

Программа читает все входные символы до точки один раз, подсчитывая в массиве, хранящем 10 целых чисел, количество каждой из цифр. Сами входные символы при этом не запоминаются. Затем проверяется — сколько в этом массиве нечетных элементов. Если больше одного, то задача решения не имеет. При наличии решения сначала печатается половина имеющихся цифр 9 (если таковые имеются, в случае нечетного числа цифр – меньшая половина), затем 8 и т.д. до 0, потом печатается цифра, которая встречается во входных данных нечетное число раз, а затем – оставшаяся половина цифр 0 (если таковые имеются, в случае нечетного числа цифр – меньшая половина), 1, и т.д. до 9. Если никаких цифр, кроме 0, во входных данных нет, то задача имеет решение, только если этот ноль единственный. Если нулей четное число, а ненулевая цифра единственная, то решения не существует.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, для строк, состоящих не более чем из 255 символов), или которая только умеет определять, имеет ли задача решение.

Пример правильной и эффективной программы на языке Паскаль:

```
var a:array['0'..'9'] of integer;
    c, c_odd: char;
    i, k: integer;
    f: boolean;
begin
  for c:='0' to '9' do a[c]:=0;
  read(c);
  while c<>'.' do
    begin
      if c in ['0' .. '9'] then a[c] := a[c] + 1;
      read(c);
    end;
  k := 0; {количество цифр, встречающихся нечетное число раз}
  for c := '0' to '9' do
    if a[c] mod 2 = 1 then
      begin
        k := k + 1;
        c_odd := c
      end;
  f := (a['0'] = 1);
  for c := '1' to '9' do
    if (a[c] > 1) or (a[c] = 1) and (a['0'] = 0) then f := true;
  if (k > 1)or not f then writeln('NO') else
  begin
    writeln('YES');
    for c := '9' downto '0' do
      for i := 1 to a[c] div 2 do
        write(c);
    if k = 1 then
      write(c_odd);
    for c := '0' to '9' do
```

```
      for i := 1 to a[c] div 2 do
        write(c);
  end
end.
```

Пример правильной и эффективной программы на языке Бейсик:

```
DIM k, i, j, iodd, a(9) AS INTEGER
FOR i = 0 TO 9
a(i) = 0
NEXT
INPUT c$
DO WHILE NOT (c$ = ".")
  IF c$ >= "0" AND c$ <= "9" THEN
   a(ASC(c$) - ASC("0")) = a(ASC(c$) - ASC("0")) + 1
  ENDIF
  INPUT c$
LOOP
k = 0
IF a(0) = 1 THEN f = 1 ELSE f = 0
FOR i = 0 TO 9
  IF a(i) MOD 2 = 1 THEN
    k = k + 1
    iodd = i
  END IF
  IF i > 0 AND a(i) > 1 THEN f = 1
NEXT
IF k = 1 AND a(0) = 0 THEN f = 1
IF k > 1 OR f = 0 THEN
  PRINT "NO"
  END
ENDIF
PRINT "YES"
FOR i = 9 TO 0 STEP -1
    FOR j = 1 TO a(i) \ 2
      PRINT i;
    NEXT
NEXT
IF k = 1 THEN PRINT iodd;
FOR i = 0 TO 9
  FOR j = 1 TO a(i) \ 2
     PRINT i;
  NEXT
NEXT
END
```

Указания по оцениванию	Баллы
Программа работает верно, т.е. определяет, имеет ли задача решение для любых входных данных произвольного размера, и строит максимальное искомое число, не сохраняя входные данные в строке или массиве символов. Программа просматривает входные данные один раз, в тексте программы не анализируется каждая цифра в отдельности. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для	4

соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).	
Программа работает верно, но входные данные запоминаются в массиве символов или строке, или входные данные считываются несколько раз. Возможно, каждая цифра обрабатывается явным образом (10 операторов IF, в том числе с использованием многоточия, или оператор CASE, содержащий 10 вариантов). Возможно, сохраненные входные данные сортируются одним из стандартных алгоритмов сортировки путем перестановки входных символов, или ответ формируется путем перестановки входных цифр. Допускается наличие от одной до трех синтаксических ошибок, описанных выше. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, не относящаяся к алгоритму решения задачи в целом, например, ошибка в принципиально верно организованном вводе данных или в обработке числа, состоящего из одних нулей или из четного числа нулей и одной ненулевой цифры.	3
Программа работает в целом верно, эффективно или нет, но, возможно, выводит значение не максимального искомого числа. Возможно в реализации алгоритма содержатся 1–2 ошибки (используется знак “/” вместо “\”, “div” вместо “mod”, выход за границу массива, перевод символов в числа, используется знак “<” вместо “<=”, “or” вместо “and” и т.п.). Возможно некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	2
Программа, возможно, неверно работает при некоторых входных данных, например, при наличии цифры, которая встречается нечетное число раз. Возможно выводит только “NO” или “YES” и не выводит искомое число, или выводит его неверно. При использовании сортировки допущены ошибки в ее реализации. Допускается до 4 различных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок, описанных выше.	1
Задание не выполнено или выполнено неверно.	0
Максимальный балл	*4*