

Критерии оценивания заданий с развёрнутым ответом

C1 Требовалось написать программу, при выполнении которой с клавиатуры вводятся последовательность из шести неотрицательных целых чисел, не превышающих 10^6 , подсчитывается и выводится сумма введённых чётных чисел или 0, если чётных чисел в последовательности нет. Ученик написал такую программу:

Бейсик	Паскаль
<pre>DIM N, S AS LONG DIM I AS INTEGER S = 1 FOR I = 1 TO 6 INPUT N IF I MOD 2 = 0 THEN S = S + N END IF NEXT I PRINT S END</pre>	<pre>var n, s: longint; i: integer; begin s:=1; for i:=1 to 6 do begin readln(n); if i mod 2 = 0 then s := s + n; end; write(s); end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> void main(){ long int n, s; int i; s = 1; for (i=1; i<=6; ++i) { scanf("%ld",&n); if (i%2 == 0) s = s + n; } printf("%ld", s); }</pre>	<pre>алг нач цел n, s цел i s := 1 нц для i от 1 до 6 ввод n если mod(i,2) = 0 то s := s + n все кц вывод s кон</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности 1, 1, 2, 3, 5, 8.
2. Приведите пример последовательности, при вводе которой 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.

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

1. При вводе последовательности 1, 1, 2, 3, 5, 8 программа выведет число 13.
Комментарий для экспертов. Приведённая программа вместо подсчёта суммы чётных чисел считает сумму чисел, стоящих на чётных позициях в последовательности, и дополнительно увеличивает эту сумму на 1 из-за неверной инициализации.

2. Пример последовательности, для которой программа даёт верный ответ: 2, 1, 1, 2, 1, 2.

Комментарий для экспертов. При любом вводе данных результатом программы будет сумма второго, четвертого и шестого чисел, увеличенная на 1. Чтобы результат был верным, нужно подобрать такую последовательность, в которой эта сумма совпадет с суммой чётных чисел. Например, в приведённой выше последовательности обе суммы равны 6.

3. Ошибки содержатся в двух строках программы:

1) неверная инициализация переменной s: вместо значения 1 нужно присваивать 0;

2) неверная проверка условия: вместо чётности i нужно проверять чётность n.

Пример исправления для языка Паскаль

Первая строка с ошибкой:

```
s := 1;
```

Исправленная строка:

```
s := 0;
```

Вторая строка с ошибкой:

```
if i mod 2 = 0 then
```

Исправленная строка:

```
if n mod 2 = 0 then
```

В программах на других языках ошибочные строки и их исправления аналогичны.

Допустимы избыточные скобки, не изменяющие правильный порядок действий. Незначительной опiskeй, не влияющей на оценку, следует считать отсутствие слова then при исправлении второй ошибки в программах на Бейсике и Паскале.

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1. Указать результат программы при данном вводе. Это действие считается выполненным, если указан верный результат работы программы при заданных входных данных. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать пример ввода, при котором программа выводит верный ответ. Это действие считается выполненным, если указан пример ввода, состоящий ровно из шести неотрицательных целых чисел, при котором программа выдаёт верный ответ. В сложных случаях можно ввести текст программы в компьютер и исполнить для предложенного экзаменуемым варианта ввода.</p> <p>Экзаменуемый не обязан описывать все ситуации, в которых программа выдаёт верный ответ, ему достаточно указать пример ввода, при котором это происходит.</p> <p>Если экзаменуемый приводит несколько примеров, действие считается выполненным только в том случае, если программа выдаёт верный ответ для всех приведённых примеров.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе строки с ошибками и предложены верные варианты исправления и при этом никакие верные строки программы не указаны в качестве неверных. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).</p>	
<p>Правильно выполнены все действия: – указан верный результат для приведённого примера входных данных; – дан пример последовательности, для которой программа с ошибками выдаёт верный результат; – указаны и исправлены две ошибки в программе; – не указаны в качестве ошибочных никакие другие строки программы.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом), найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.</p> <p>2. Выполнены два первых действия (верный результат при</p>	2

<p>указанных данных, верный пример последовательности с правильным результатом), найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.</p> <p>3. Выполнено одно из первых двух действий (верный результат при указанных данных или верный пример последовательности с правильным результатом), найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной.</p>	
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом). При этом несущественно, насколько правильно выполнено третье действие.</p> <p>2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2.</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.</p>	0
<i>Максимальный балл</i>	3

C2

Дан массив, содержащий 2014 вещественных чисел. Напишите на одном из языков программирования программу, находящую в этом массиве два соседних элемента, значения которых наиболее близки, то есть абсолютная величина их разности минимальна. Если таких пар несколько, можно взять любую из них. Программа должна вывести найденные элементы. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

Бейсик	Паскаль
<pre>N=2014 DIM A(N) AS SINGLE DIM D, R AS SINGLE DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N=2014; var a: array [1..N] of real; d, r: real; i, j, k: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>

Си	Алгоритмический язык
<pre>#include <stdio.h> #include <math.h> #define N 2014 void main(){ float a[N]; float d, r; int i, j, k; for (i=0; i<N; i++) scanf("%f", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N=2014 <u>вещтаб</u> a[1:N] <u>вещ</u> d, r; <u>цел</u> i, j, k <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.4). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа рассматривает все пары соседних чисел в массиве, определяет абсолютное значение разности элементов в каждой паре и находит минимум среди этих разностей. Для запоминания и последующего вывода можно хранить индекс первого элемента текущей наиболее подходящей пары, а в конце программы выводить элемент с данным индексом и следующий за ним.

Пример программы на Паскале

```
r:=abs(a[2]-a[1]);
k:=1;
for i:=2 to N-1 do begin
    d:=abs(a[i+1]-a[i]);
    if d<r then begin
        r:=d;
        k:=i;
    end;
end;
writeln(a[k], ' ', a[k+1]);
```

Можно запоминать индексы обоих элементов пары.

Пример программы на Си	
<pre>r=fabs(a[1]-a[0]); j=1; k=2; for (i=1; i<N-1; ++i) { d=fabs(a[i+1]-a[i]); if (d<r) { r=d; j=i; k=i+1; } } printf("%f %f\n", a[j], a[k]);</pre>	
Можно хранить не индексы, а значения элементов. Но поскольку разрешено использование только двух вещественных переменных, в этом случае не удастся запоминать разности, их придётся каждый раз вычислять заново.	
Пример программы на Алгоритмическом языке	
<pre>d:=a[1]; r:=a[2] <u>нц для</u> i <u>от</u> 2 <u>до</u> N-1 <u>если</u> abs(a[i+1]-a[i]) < abs(r-d) <u>то</u> d:=a[i] r:=a[i+1] <u>все</u> <u>кц</u> <u>вывод</u> d, " ", r</pre>	
Допускаются и другие способы решения, если они соответствуют указанным в условии ограничениям и приводят к правильному ответу.	
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 языках, использованных в задании. 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. В частности, оценка не снижается за использование не соответствующей типу данных функции вычисления модуля (например, abs вместо fabs в программе на Си).	2

Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку). 1. Отсутствие инициализации или неверная инициализация. Например, в программе, запоминающей индексы, нет начального присваивания индексам. 2. Неверно определены границы цикла проверки, в результате проверяются не все пары или происходит выход за границы массива. 3. Вместо абсолютного значения разности используется просто разность элементов. 4. При вычислении разности вместо элемента используется его абсолютное значение. 5. Некорректно используются типы данных. Например, для хранения вещественного значения используется переменная целого типа. 6. Отсутствует вывод ответа. 7. Используется переменная, не объявленная в разделе описания переменных. 8. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	1
Не выполнены условия, позволяющие поставить 1 или 2 балла.	0
<i>Максимальный балл</i>	2

С3

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 43. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 43 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 42$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

- а) При каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения.
б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.
- Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём
– Петя не может выиграть за один ход, но
– Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.
Для указанного значения S опишите выигрышную стратегию Пети.
- Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.
Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

- а) Петя может выиграть, если $S = 22, \dots, 42$, удвоив количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 42 камней.
б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 21$ камень. Тогда после первого хода Пети в куче будет 22 камня, или 23 камня, или 42 камня. Во всех случаях Ваня удваивает количество камней и выигрывает первым ходом.
- Возможные значения S : 19, 20. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 21 камня. Эта позиция разобрана п. 1 б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.
- Возможное значение S : 18. После первого хода Пети в куче будет 19, 20 или 38 камней. Если в куче станет 38 камней, Ваня удвоит количество камней и выигрывает первым ходом. Ситуации, когда в куче 19 или 20 камней, уже разобраны в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
18	18+1=19	19+2=21	21+1=22	<u>22*2=44</u>
			21+2=23	<u>23*2=46</u>
			21*2=42	<u>42*2=84</u>
	18+2=20	20+1=21	21+1=22	<u>22*2=44</u>
			21+2=23	<u>23*2=46</u>
			21*2=42	<u>42*2=84</u>
18*2=36	36*2=72			

- В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

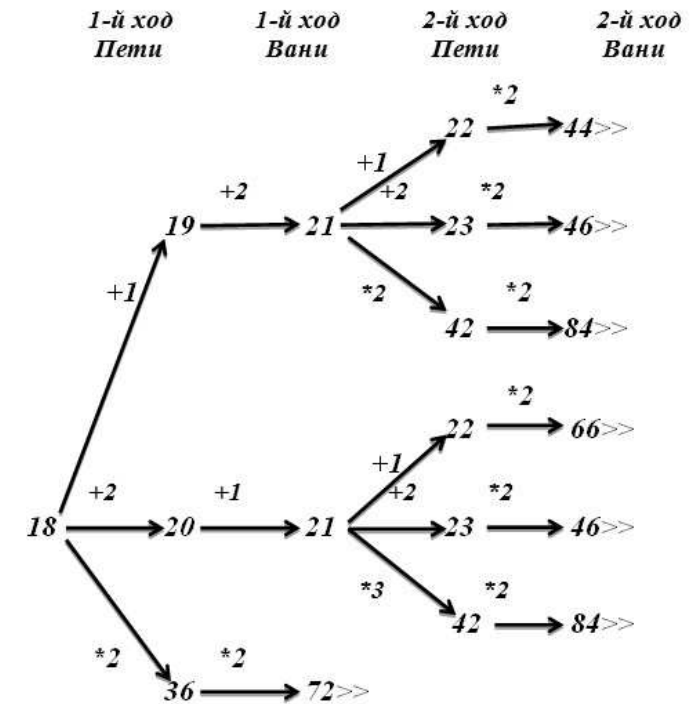


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию**Баллы**

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.

Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, т. е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.

<p>Первое задание считается выполненным частично, если (1) правильно указаны все позиции, в которых Петя выигрывает первым ходом, (2) правильно указана позиция, в которой Ваня выигрывает первым ходом, и (3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней.</p> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например, с помощью дерева всех партий, возможных при выбранной стратегии Пети.</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. В этом дереве в каждой позиции, где должен ходить Петя, должны быть разобраны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий стратегии, которую выбрал Ваня.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.</p>	
<p>Выполнены второе и третье задания. Первое задание выполнено полностью или частично.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Задание 3 выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S. 	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Первое задание выполнено полностью. 2. Во втором задании правильно указано одно из двух возможных значений S и для этого значения указана и обоснована выигрышная стратегия Пети. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение S. 4. Для второго и третьего задания правильно указаны значения S. 	1
<p>Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл.</p>	0
<i>Максимальный балл</i>	3

C4

На плоскости дан набор точек с целочисленными координатами. Необходимо найти четырёхугольник наибольшей площади с вершинами в этих точках, две вершины которого лежат на оси Ox , а две оставшиеся – по разные стороны от оси Ox .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа: сначала координата x , затем координата y очередной точки.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь четырёхугольника, удовлетворяющего условиям задачи. Если такого четырёхугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
6
0 0
2 0
0 2
3 -3
5 -5
6 6
```

Пример выходных данных для приведённого выше примера входных данных:

```
11
```

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Искомый четырёхугольник состоит из двух треугольников с общим основанием, лежащим на оси Ox, при этом один треугольник лежит выше этой оси, другой – ниже. Площадь четырёхугольника будет максимальной, если вершины на оси Ox будут расположены как можно дальше друг от друга, а вершины, не лежащие на этой оси, – как можно дальше от неё.</p> <p>Программа читает исходные данные, не запоминая все точки в массиве. Для каждой точки проверяется её принадлежность оси Ox (условие $y=0$). Среди точек, лежащих на оси, необходимо найти наиболее далеко отстоящие друг от друга – они дадут наибольшее возможное общее основание двух треугольников. Это будут точки с наименьшим и наибольшим значением координаты x. Среди точек, не лежащих на оси Ox, надо найти две точки, расположенные по разные стороны от оси и как</p>

можно дальше от неё, – они дадут наибольшие возможные значения высот треугольников. Это будут точки с наибольшим положительным и наименьшим отрицательным значением координаты y .

Таким образом, задача сводится к нахождению максимального и минимального x среди точек, y которых $y=0$, максимального и минимального y среди остальных точек и нахождению площади четырёхугольника на основе этих данных.

Перед выводом результата необходимо убедиться в существовании искомого четырёхугольника.

Пример правильной и эффективной программы на языке Паскаль

```
program c4;
var
  n: integer;
  x, y: integer;
  xmin, xmax: integer;
  xsearch: boolean;
  ymin, ymax: integer;
  i: integer;
  s: real;

begin
  xsearch := true;
  xmin := 0; xmax := 0;
  ymin:=0; ymax := 0;
  readln(n);
  for i:=1 to n do begin
    readln(x,y);
    if y=0 then begin
      if xsearch or (x<xmin) then xmin:=x;
      if xsearch or (x>xmax) then xmax:=x;
      xsearch:=false;
    end
    else if y<ymin then ymin:=y
    else if y>ymax then ymax:=y
  end;
  if (xmax>xmin) and (ymin<0) and (ymax>0)
  then s := (xmax-xmin)*(ymax-ymin)/2
  else s := 0;
  writeln(s);
end.
```

Пример правильной и эффективной программы на языке Бейсик

```
DIM n AS INTEGER
DIM x, y AS INTEGER
DIM xmin, xmax AS INTEGER
DIM xsearch AS INTEGER
DIM ymin, ymax AS INTEGER
DIM i AS INTEGER
DIM s AS DOUBLE

xsearch = 1
xmin = 0: xmax = 0
ymin = 0: ymax = 0
INPUT n
FOR i = 1 TO n
  INPUT x, y
  IF y = 0 THEN
    IF xsearch = 1 OR x < xmin THEN xmin = x
    IF xsearch = 1 OR x > xmax THEN xmax = x
    xsearch = 0
  ELSEIF y < ymin THEN ymin = y
  ELSEIF y > ymax THEN ymax = y
  END IF
NEXT i
IF xmax > xmin AND ymin < 0 AND ymax > 0 THEN
  s = (xmax - xmin) * (ymax - ymin) / 2
ELSE
  s = 0
END IF
PRINT s
```

Пример правильной и эффективной программы на Алгоритмическом языке

```

алг с4
нач
  цел n
  цел x, y
  цел xmin=0, xmax=0
  лог xsearch=да
  цел ymin=0, ymax=0
  цел i
  вещ s

  ввод n
  нц для i от 1 до n
    ввод x, y
    если y=0
      то
        если xsearch или x<xmin то xmin:=x все
        если xsearch или x>xmax то xmax:=x все
        xsearch:=нет
      иначе
        если y<ymin то ymin:=y все
        если y>ymax то ymax:=y все
    все
  кц

  если xmax > xmin и ymin < 0 и ymax > 0
    то s:=(xmax-xmin)*(ymax-ymin)/2
    иначе s:=0
  все
  вывод s
кон

```

Указания по оцениванию**Баллы**

Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. 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 соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).

4

Не выполнены условия, позволяющие поставить 4 балла. Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой

3

структуре данных, размер которой соответствует количеству точек. При этом обработка данных происходит с использованием эффективного по времени алгоритма, аналогичного приведённым выше.

Допускается одна из следующих ошибок.

1. Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать.
2. Перепутаны координаты x и y , например, ищутся максимальное и минимальное значения y при $x = 0$.
3. При вычислении площади нижнего треугольника не используется модуль, в результате его площадь учитывается со знаком «минус».
4. Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула.
5. Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (`div` в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа или имеются другие подобные ошибки, приводящие к неверному результату при дробном ответе.
6. Неверно обрабатывается ситуация, когда искомым четырёхугольник отсутствует.

Допускается наличие от одной до трёх синтаксических ошибок, описанных выше.

Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие четырёхугольники или составляющие треугольники, вычисляется и сравнивается их площадь.

В реализации алгоритма допущено более одной ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях.

Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.

2

Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Программа работает в отдельных частных случаях. Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл</i>	<i>4</i>

Критерии оценивания заданий с развёрнутым ответом

C1 Требовалось написать программу, при выполнении которой с клавиатуры вводится последовательность из шести неотрицательных целых чисел, не превышающих 10^6 , подсчитывается и выводится сумма введённых нечётных чисел или 0, если нечётных чисел в последовательности нет. Ученик написал такую программу:

Бейсик	Паскаль
<pre>DIM N, S AS LONG DIM I AS INTEGER S = 1 FOR I = 1 TO 6 INPUT N IF I MOD 2 = 1 THEN S = S + N END IF NEXT I PRINT S END</pre>	<pre>var n, s: longint; i: integer; begin s:=1; for i:=1 to 6 do begin readln(n); if i mod 2 = 1 then s := s + n; end; write(s); end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> void main(){ long int n, s; int i; s = 1; for (i=1; i<=6; ++i) { scanf("%ld",&n); if (i%2 == 1) s = s + n; } printf("%ld", s); }</pre>	<pre>алг нач цел n, s цел i s := 1 нц для i от 1 до 6 ввод n если mod(i,2) = 1 то s := s + n все кц вывод s кон</pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности 1, 1, 2, 3, 5, 8.
2. Приведите пример последовательности, при вводе которой 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.

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

1. При вводе последовательности 1, 1, 2, 3, 5, 8 программа выведет число 9.
Комментарий для экспертов. Приведённая программа вместо подсчёта суммы нечётных чисел считает сумму чисел, стоящих на нечётных позициях в последовательности, и дополнительно увеличивает эту сумму на 1 из-за неверной инициализации.

2. Пример последовательности, для которой программа даёт верный ответ: 2, 3, 1, 2, 1, 2.

Комментарий для экспертов. При любом вводе данных результатом программы будет сумма первого, третьего и пятого чисел, увеличенная на 1. Чтобы результат был верным, нужно подобрать такую последовательность, в которой эта сумма совпадет с суммой нечётных чисел. Например, в приведённой выше последовательности обе суммы равны 5.

3. Ошибки содержатся в двух строках программы:

1) неверная инициализация переменной s: вместо значения 1 нужно присваивать 0;

2) неверная проверка условия: вместо нечётности i нужно проверять нечётность n.

Пример исправления для языка Паскаль

Первая строка с ошибкой:

```
s:=1;
```

Исправленная строка:

```
s:=0;
```

Вторая строка с ошибкой:

```
if i mod 2 = 1 then
```

Исправленная строка:

```
if n mod 2 = 1 then
```

В программах на других языках ошибочные строки и их исправления аналогичны.

Допустимы избыточные скобки, не изменяющие правильный порядок действий. Незначительной опiskой, не влияющей на оценку, следует считать отсутствие слова then при исправлении второй ошибки в программах на Бейсике и Паскале.

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1. Указать результат программы при данном вводе. Это действие считается выполненным, если указан верный результат работы программы при заданных входных данных. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать пример ввода, при котором программа выводит верный ответ. Это действие считается выполненным, если указан пример ввода, состоящий ровно из шести неотрицательных целых чисел, при котором программа выдаёт верный ответ. В сложных случаях можно ввести текст программы в компьютер и исполнить для предложенного экзаменуемым варианта ввода.</p> <p>Экзаменуемый не обязан описывать все ситуации, в которых программа выдаёт верный ответ, ему достаточно указать пример ввода, при котором это происходит.</p> <p>Если экзаменуемый приводит несколько примеров, действие считается выполненным только в том случае, если программа выдаёт верный ответ для всех приведённых примеров.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе строки с ошибками и предложены верные варианты исправления и при этом никакие верные строки программы не указаны в качестве неверных. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).</p>	
<p>Правильно выполнены все действия:</p> <ul style="list-style-type: none"> – указан верный результат для приведённого примера входных данных; – дан пример последовательности, для которой программа с ошибками выдаёт верный результат; – указаны и исправлены две ошибки в программе; – не указаны в качестве ошибочных никакие другие строки программы. 	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом), найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.</p> <p>2. Выполнены два первых действия (верный результат при</p>	2

<p>указанных данных, верный пример последовательности с правильным результатом), найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.</p> <p>3. Выполнено одно из первых двух действий (верный результат при указанных данных или верный пример последовательности с правильным результатом), найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной.</p>	
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом). При этом несущественно, насколько правильно выполнено третье действие.</p> <p>2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2.</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.</p>	0
<i>Максимальный балл</i>	<i>3</i>

C2

Дан массив, содержащий 2014 вещественных чисел. Напишите на одном из языков программирования программу, находящую в этом массиве два соседних элемента, значения которых наименее близки, то есть абсолютная величина их разности максимальна. Если таких пар несколько, можно взять любую из них. Программа должна вывести найденные элементы. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

Бейсик	Паскаль
<pre>N=2014 DIM A(N) AS SINGLE DIM D, R AS SINGLE DIM I, J, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N=2014; var a: array [1..N] of real; d, r: real; i, j, k: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>

Си	Алгоритмический язык
<pre>#include <stdio.h> #include <math.h> #define N 2014 void main(){ float a[N]; float d, r; int i, j, k; for (i=0; i<N; i++) scanf("%f", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N=2014 <u>вещтаб</u> a[1:N] <u>вещ</u> d, r; <u>цел</u> i, j, k <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.4). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа рассматривает все пары соседних чисел в массиве, определяет абсолютное значение разности элементов в каждой паре и находит максимум среди этих разностей. Для запоминания и последующего вывода можно хранить индекс первого элемента текущей наиболее подходящей пары, а в конце программы выводить элемент с данным индексом и следующий за ним.

Пример программы на Паскале

```
r:=abs(a[2]-a[1]);
k:=1;
for i:=2 to N-1 do begin
    d:=abs(a[i+1]-a[i]);
    if d>r then begin
        r:=d;
        k:=i;
    end;
end;
writeln(a[k], ' ', a[k+1]);
```

В качестве начального значения для максимальной разности можно использовать ноль, но даже в этом случае нужно использовать стартовое значение индекса. Оно будет использовано, если все элементы в массиве окажутся одинаковыми.

Можно запоминать не один, а оба индекса найденной пары.

Пример программы на Си
<pre>r=0; j=0; k=1; for (i=0; i<N-1; ++i) { d=fabs(a[i+1]-a[i]); if (d>r) { r=d; j=i; k=i+1; } } printf("%f %f\n", a[j], a[k]);</pre>
<p>Можно хранить не индексы, а значения элементов. Но поскольку разрешено использование только двух вещественных переменных, в этом случае не удастся запоминать разности, их придётся каждый раз вычислять заново.</p>
Пример программы на Алгоритмическом языке
<pre>d:=a[1]; r:=a[2] нц для i от 2 до N-1 если abs(a[i+1]-a[i]) > abs(r-d) то d:=a[i] r:=a[i+1] все кц вывод d, " ", r</pre>
<p>Допускаются и другие способы решения, если они соответствуют указанным в условии ограничениям и приводят к правильному ответу.</p>

Указания по оцениванию	Баллы
<p>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 языках, использованных в задании. 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. В частности, оценка не снижается за использование не соответствующей типу данных функции вычисления модуля (например, abs вместо fabs в программе на Си).</p>	2

Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку): 1. Отсутствие инициализации или неверная инициализация. Например, в программе, запоминающей индексы, нет начального присваивания индексам. 2. Неверно определены границы цикла проверки, в результате проверяются не все пары или происходит выход за границы массива. 3. Вместо абсолютного значения разности используется просто разность элементов. 4. При вычислении разности вместо элемента используется его абсолютное значение. 5. Некорректно используются типы данных. Например, для хранения вещественного значения используется переменная целого типа. 6. Отсутствует вывод ответа. 7. Используется переменная, не объявленная в разделе описания переменных. 8. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	1
Не выполнены условия, позволяющие поставить 1 или 2 балла.	0
<i>Максимальный балл</i>	2

С3

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 47. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 47 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 46$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

- а) При каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения.
б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.
- Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём
– Петя не может выиграть за один ход, но
– Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.
Для указанного значения S опишите выигрышную стратегию Пети.
- Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.
Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

- а) Петя может выиграть, если $S = 24, \dots, 46$, удвоив количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 46 камней.
б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 23$ камня. Тогда после первого хода Пети в куче будет 24 камня, или 25 камней, или 46 камней. Во всех случаях Ваня удваивает количество камней и выигрывает первым ходом.
- Возможные значения S : 21, 22. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 23 камней. Эта позиция разобрана п. 1 б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выиграет.
- Возможное значение S : 20. После первого хода Пети в куче будет 21, 22 или 40 камней. Если в куче станет 40 камней, Ваня удвоит количество камней и выиграет первым ходом. Ситуации, когда в куче 21 или 22 камня, уже разобраны в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
20	20+1=21	21+2=23	23+1=24	<u>24*2=48</u>
			23+2=25	<u>25*2=50</u>
			23*2=46	<u>46*2=92</u>
	20+2=22	22+1=23	23+1=24	<u>24*2=48</u>
			23+2=25	<u>25*2=50</u>
			23*2=46	<u>46*2=92</u>
20*2=40		<u>40*2=80</u>		

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

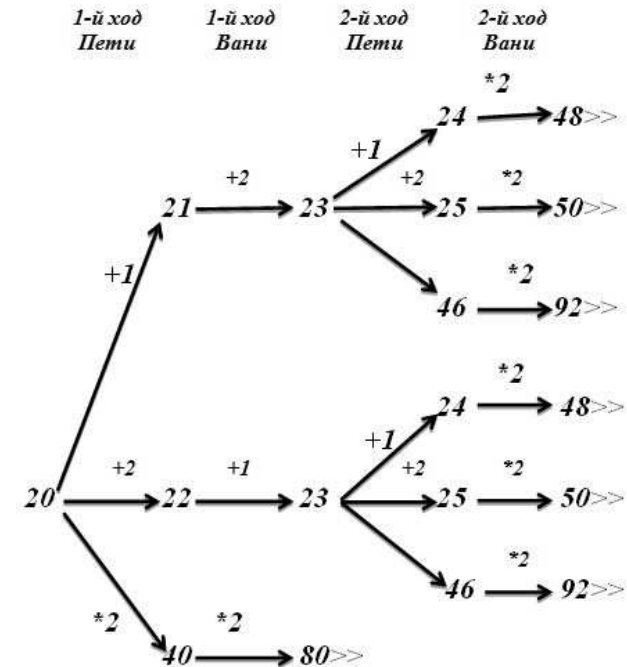


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию**Баллы**

В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.

Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, т. е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.

<p>Первое задание считается выполненным частично, если (1) правильно указаны все позиции, в которых Петя выигрывает первым ходом, (2) правильно указана позиция, в которой Ваня выигрывает первым ходом, и (3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней.</p> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех партий, возможных при выбранной стратегии Пети.</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. В этом дереве в каждой позиции, где должен ходить Петя, должны быть разобраны все возможные ходы, а для позиций, где должен ходить Ваня, – только ход, соответствующий стратегии, которую выбрал Ваня.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.</p>	
<p>Выполнены второе и третье задания. Первое задание выполнено полностью или частично.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу.</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Задание 3 выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S. 	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Первое задание выполнено полностью. 2. Во втором задании правильно указано одно из двух возможных значений S и для этого значения указана и обоснована выигрышная стратегия Пети. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указано значение S. 4. Для второго и третьего задания правильно указаны значения S. 	1
<p>Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл.</p>	0
<i>Максимальный балл</i>	3

С4

На плоскости дан набор точек с целочисленными координатами. Необходимо найти четырёхугольник наибольшей площади с вершинами в этих точках, две вершины которого лежат на оси Oy , а две оставшиеся – по разные стороны от оси Oy .

Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу. Перед текстом программы кратко опишите алгоритм решения задачи и укажите используемый язык программирования и его версию.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек N .

Каждая из следующих N строк содержит два целых числа: сначала координата x , затем координата y очередной точки.

Описание выходных данных

Программа должна вывести одно число – максимальную площадь четырёхугольника, удовлетворяющего условиям задачи. Если такого четырёхугольника не существует, программа должна вывести ноль.

Пример входных данных:

```
6
0 0
2 0
0 2
3 -3
-5 -5
6 6
```

Пример выходных данных для приведённого выше примера входных данных:

```
11
```

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

Искомый четырёхугольник состоит из двух треугольников с общим основанием, лежащим на оси Oy , при этом один треугольник лежит выше этой оси, другой – ниже. Площадь четырёхугольника будет максимальной, если вершины на оси Oy будут расположены как можно дальше друг от друга, а вершины, не лежащие на этой оси, – как можно дальше от неё.

Программа читает исходные данные, не запоминая все точки в массиве. Для каждой точки проверяется её принадлежность оси Oy (условие $x=0$). Среди точек, лежащих на оси, необходимо найти наиболее далеко отстоящие друг от друга – они дадут наибольшее возможное общее основание двух треугольников. Это будут точки с наименьшим и наибольшим значением координаты y . Среди точек, не лежащих на оси Oy , надо найти две точки, расположенные по разные стороны от оси и как можно дальше от неё, – они дадут наибольшие возможные значения высот треугольников. Это будут точки с наибольшим положительным и наименьшим отрицательным значением координаты x .

Таким образом, задача сводится к нахождению максимального и минимального y среди точек, у которых $x=0$, максимального и минимального x среди остальных точек и нахождению площади четырёхугольника на основе этих данных.

Перед выводом результата необходимо убедиться в существовании искомого четырёхугольника.

Пример правильной и эффективной программы на языке Паскаль

```

program c4;
var
  n: integer;
  x, y: integer;
  ymin, ymax: integer;
  ysearch: boolean;
  xmin, xmax: integer;
  i: integer;
  s: real;

begin
  ysearch := true;
  ymin:=0; ymax := 0;
  xmin := 0; xmax := 0;
  readln(n);
  for i:=1 to n do begin
    readln(x,y);
    if x=0 then begin
      if ysearch or (y<ymin) then ymin:=y;
      if ysearch or (y>ymax) then ymax:=y;
      ysearch:=false;
    end
  end

```

```

  else if x<xmin then xmin:=x
  else if x>xmax then xmax:=x
end;
if (ymax>ymin) and (xmin<0) and (xmax>0)
  then s := (xmax-xmin)*(ymax-ymin)/2
  else s := 0;
writeln(s);
end.

```

Пример правильной и эффективной программы на языке Бейсик

```

DIM n AS INTEGER
DIM x, y AS INTEGER
DIM ymin, ymax AS INTEGER
DIM ysearch AS INTEGER
DIM xmin, xmax AS INTEGER
DIM i AS INTEGER
DIM s AS DOUBLE

ysearch = 1
ymin = 0: ymax = 0
xmin = 0: xmax = 0
INPUT n
FOR i = 1 TO n
  INPUT x, y
  IF x = 0 THEN
    IF ysearch = 1 OR y < ymin THEN ymin = y
    IF ysearch = 1 OR y > ymax THEN ymax = y
    ysearch = 0
  ELSEIF x < xmin THEN xmin = x
  ELSEIF x > xmax THEN xmax = x
  END IF
NEXT i
IF ymax > ymin AND xmin < 0 AND xmax > 0 THEN
  s = (xmax - xmin) * (ymax - ymin) / 2
ELSE
  s = 0
END IF
PRINT s

```


Пример правильной и эффективной программы на Алгоритмическом языке

```

алг с4
нач
  цел n
  цел x, y
  цел ymin=0, ymax=0
  лог ysearch=да
  цел xmin=0, xmax=0
  цел i
  вещ s

  ввод n
  нц для i от 1 до n
    ввод x, y
    если x=0
      то
        если ysearch или y<ymin то ymin:=y все
        если ysearch или y>ymax то ymax:=y все
        ysearch:=нет
      иначе
        если x<xmin то xmin:=x все
        если x>xmax то xmax:=x все
    все
  кц

  если ymax > ymin и xmin < 0 и xmax > 0
    то s:=(xmax-xmin)*(ymax-ymin)/2
    иначе s:=0
  все
  вывод s
кон

```

Указания по оцениванию	Баллы
Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. 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 соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку).	4
Не выполнены условия, позволяющие поставить 4 балла. Программа работает верно, но размер используемой памяти зависит от количества исходных данных. Например, входные данные (координаты точек) запоминаются в массиве или другой структуре данных, размер которой соответствует количеству	3

точек. При этом обработка данных происходит с использованием эффективного по времени алгоритма, аналогичного приведённым выше.

Допускается одна из следующих ошибок.

1. Поиск минимума или максимума не учитывает, что первый подходящий элемент может оказаться на любом месте в исходных данных или вообще отсутствовать.
2. Перепутаны координаты x и y , например, ищутся максимальное и минимальное значения x при $y=0$.
3. При вычислении площади нижнего треугольника не используется модуль, в результате его площадь учитывается со знаком «минус».
4. Все вершины определены правильно, но площадь треугольника определена неверно, например, использована неверная формула.
5. Не учитывается, что вычисленное значение площади может быть нецелым. Например, значение площади присваивается переменной целого типа, при вычислении площади используется операция целочисленного деления (`div` в Паскале, деление целых величин без приведения типов в Си), при форматном выводе используется формат целого числа, или имеются другие подобные ошибки, приводящие к неверному результату при дробном ответе.
6. Неверно обрабатывается ситуация, когда искомым четырёхугольник отсутствует.

Допускается наличие от одной до трёх синтаксических ошибок, описанных выше.

Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Возможны переборные решения, при которых все точки хранятся в массиве, из них выбираются подходящие четырёхугольники или составляющие треугольники, вычисляется и сравнивается их площадь. В реализации алгоритма допущено более одной ошибки из числа перечисленных в предыдущем пункте или допущены другие ошибки, приводящие к неверной работе программы в отдельных случаях. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.

2

Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Программа работает в отдельных частных случаях. Один балл также ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла.	0
<i>Максимальный балл</i>	<i>4</i>