

Критерии оценивания заданий с развёрнутым ответом**24**

Дано целое положительное число N , не превосходящее 1000. Необходимо определить, является ли это число степенью числа 3. То есть требуется определить, существует ли такое целое число K , что $3^K = N$, и вывести это число либо сообщение, что такого числа не существует.

Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, K AS INTEGER INPUT N K = 0 WHILE K MOD 3 = 0 K = K + 1 N = N \ 3 WEND IF N > 0 THEN PRINT K ELSE PRINT "Не существует" END IF END </pre>	<pre> n = int(input()) k = 0 while k%3 == 0: k = k + 1 n = n // 3 if n > 0: print(k) else: print("Не существует") </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел n, k ввод n k := 0 нц пока mod(k, 3)=0 k := k + 1 n := div(n, 3) кц если n > 0 то вывод k иначе вывод "Не существует" все кон </pre>	<pre> var n, k: integer; begin read(n); k := 0; while k mod 3 = 0 do begin k := k + 1; n := n div 3; end; if n > 0 then writeln(k) else writeln('Не существует') end. </pre>

Си

```
#include <stdio.h>
int main(){
    int n, k;
    scanf("%d",&n);
    k = 0;
    while (k%3 == 0) {
        k = k + 1;
        n = n / 3;
    }
    if (n > 0)
        printf("%d", k);
    else
        printf("Не существует");
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 9.
2. Приведите пример числа, при вводе которого приведённая программа напечатает то, что требуется.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

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)	
<p>1. При вводе числа 9 программа выведет число 1.</p> <p>2. Примеры чисел, при вводе которых программа выводит корректный ответ: 2, 3. Других чисел нет.</p> <p>Комментарий для экспертов. После выполнения программы при любом введённом n значение k будет равно 1 (тело цикла выполнится ровно 1 раз). В результате программа напечатает либо 1 (если $n \geq 3$), либо «Не существует» (в противном случае). Таким образом, программа выводит корректный ответ, только если введено 2 или 3. Экзаменуемому достаточно указать любое из этих чисел. Отметим, что при $n=1$ программа напечатает «Не существует», что неверно (должно быть напечатано «0»).</p> <p>3. Программа содержит две ошибки:</p> <ol style="list-style-type: none"> 1) неверное условие цикла; 2) неверное условие при печати результата. <p>Пример исправления для языка Паскаль:</p> <p>Первая ошибка:</p> <pre>while k mod 3 = 0 do begin</pre> <p>Исправленная строка:</p> <pre>while n mod 3 = 0 do begin</pre> <p>Вторая ошибка:</p> <pre>if n>0 then</pre> <p>Исправленная строка:</p> <pre>if n=1 then</pre> <p>Пояснение для эксперта</p> <p>После исправления первой ошибки в результате выполнения цикла значение переменной n будет равно $n_0/(3^k)$, где n_0 – введённое пользователем значение; k – максимальный показатель степени, при котором 3^k является делителем числа n_0. Число n_0 является степенью числа 3, если $n_0 = 3^k$, т.е. $n_0/(3^k) = 1$.</p> <p>В программах на других языках ошибочные строки и их исправления аналогичны.</p> <p>Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления</p>	
Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1. Указать результат программы при данном вводе.</p> <p>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 числа, при вводе которого выводится верное сообщение (верный показатель степени или текст «Не существует», если введённое число не является степенью). Ученик не обязан указывать, что будет выведено, и объяснять, как работает программа.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе ошибки и предложены верные варианты исправления, 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). Ошибка считается исправленной, если выполнены оба следующих условия:</p> <ul style="list-style-type: none"> а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа 	
Выполнены все три необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций.</p> <p>1. Выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.</p> <p>2. Выполнены два первых действия, найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.</p> <p>3. Выполнено одно из первых двух действий, найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной</p>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия. 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> <p>3. Выполнено одно из двух первых действий. Исправлена одна из двух ошибок. Не более чем одна верная строка названа ошибочной</p>	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

25

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых десятичная запись хотя бы одного числа оканчивается на 2. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 16 3 142 55 22 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Python
<pre>CONST N = 40 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>//допускается также использовать //две целочисленные переменные j и k a = [] n = 40 for i in range(0, n): a.append(int(input())) ... </pre>
Алгоритмический язык	Паскаль
<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 40 <u>целтаб</u> a[1:N] <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>	<pre>const N = 40; var a: array [1..N] of integer; i, j, k: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>
Си	Естественный язык
<pre>#include <stdio.h> #define N 40 int main() { int a[N]; int i, j, k; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... return 0; }</pre>	<p>Объявляем массив A из 40 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 40 вводим элементы массива A с 1-го по 40-й.</p> <p>...</p>

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, Free Pascal 2.6) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки решений, приводящие к правильному результату)	
На языке Паскаль	
<pre> k := 0; for i := 1 to N - 1 do if (a[i] mod 10 = 2) or (a[i + 1] mod 10 = 2) then inc(k); writeln(k); </pre>	
На алгоритмическом языке	
<pre> k := 0; нц для i от 1 до N-1 если mod(a[i],10)=2 или mod(a[i+1],10)=2 то k := k+1 все кц вывод k </pre>	
На языке Бейсик	
<pre> K = 0 FOR I = 1 TO N - 1 IF (A(I) MOD 10 = 2) OR (A(I + 1) MOD 10 = 2) THEN K = K + 1 END IF NEXT I PRINT K </pre>	
На языке Си	
<pre> k = 0; for (i = 0; i < N - 1; i++) if (a[i] % 10 == 2 a[i + 1] % 10 == 2) k++; printf("%d", k); </pre>	
На языке Python	
<pre> k = 0 for i in range(0, n - 1): if (a[i] % 10 == 2 or a[i + 1] % 10 == 2): k += 1 print(k) </pre>	

На естественном языке	
<p>Записываем в переменную K начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления текущего и следующего элементов массива на 10. Если первый или второй из полученных остатков равен 2, увеличиваем переменную K на единицу. После завершения цикла выводим значение переменной K</p>	
Указания по оцениванию	Баллы
<p><i>Общие указания</i></p> <p>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</p> <p>2. Эффективность алгоритма не имеет значения и не оценивается.</p> <p>3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. 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</p>	
Предложен правильный алгоритм, выдающий в качестве результата верное значение	2

<p>Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее <u>не более одной</u> ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1) в цикле происходит выход за границу массива (например, при использовании цикла от 1 до N); 2) не инициализируется или неверно инициализируется счётчик количества найденных пар; 3) счётчик количества пар в цикле не изменяется или изменяется неверно; 4) неверно выделяется последняя цифра числа; 5) при проверке выполнения условия для пары элементов используются неверные индексы; 6) последняя цифра выделяется не у самих элементов массива, а у их индексов; 7) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении); 8) отсутствует вывод ответа; 9) используется переменная, не объявленная в разделе описания переменных; 10) не указано или неверно указано условие завершения цикла; 11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 12) неверно расставлены операторные скобки 	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

26

Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один** камень или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 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 20. Если при этом в куче оказалось не более 30 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 17 камней и Паша удвоит количество камней в куче, то игра закончится и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 19$.

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.

1. а) При каких значениях числа S Паша может выиграть в один ход?
Укажите все такие значения и соответствующие ходы Паши.
б) У кого из игроков есть выигрышная стратегия при $S = 18, 17, 16$?
Опишите выигрышные стратегии для этих случаев.
2. У кого из игроков есть выигрышная стратегия при $S = 9, 8$? Опишите соответствующие выигрышные стратегии.
3. У кого из игроков есть выигрышная стратегия при $S = 7$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (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)

1. а) Паша может выиграть, если $S = 19$ или $S = 10, 11, 12, 13, 14, 15$. При $S = 19$ первым ходом нужно добавить в кучу один камень, при остальных указанных значениях S нужно удвоить количество камней.
 б) При $S = 16, 17$ или 18 удваивать количество камней не имеет смысла, так как после такого хода выигрывает противник. Поэтому можно считать, что единственный возможный ход – это добавление в кучу одного камня. При $S = 18$ после такого хода Паши в куче станет 19 камней. В этой позиции ходящий (т.е. Валя) выигрывает (см. п. а)): при $S = 18$ Паша (игрок, который должен ходить первым) проигрывает. Выигрышная стратегия есть у Вали.
 При $S = 17$, после того как Паша своим первым ходом добавит один камень, в куче станет 18 камней. В этой позиции ходящий (т.е. Валя) проигрывает (см. выше): при $S = 17$ Паша (игрок, который должен ходить первым) выигрывает. Выигрышная стратегия есть у Паши.
 При $S = 16$ выигрышная стратегия есть у Вали. Действительно, если Паша первым ходом удваивает количество камней, то в куче становится 32 камня, и игра сразу заканчивается выигрышем Вали. Если Паша добавляет один камень, то в куче становится 17 камней. Как мы уже знаем, в этой позиции игрок, который должен ходить (т.е. Валя), выигрывает.
 Во всех случаях выигрыш достигается тем, что при своём ходе игрок, имеющий выигрышную стратегию, должен добавить в кучу один камень.
Замечание для проверяющего. Скорее всего, решение экзаменуемого будет не столь подробным. Это не является ошибкой. Ученик может, например, нарисовать деревья всех возможных партий для указанных значений S . Другая возможность – (1) указать на то, что удваивать кучу не имеет смысла, и (2) последовательно сводить случай $S = 18$ к случаю $S = 19$, случай $S = 17$ – к случаю $S = 18$ и т.д.
 2. При $S = 9$ или 8 выигрышная стратегия есть у Паши. Она состоит в том, чтобы удвоить количество камней в куче и получить кучу, в которой будет соответственно 18 или 16 камней. В обоих случаях игрок, который будет делать ход (теперь это Валя), проигрывает (п. 1б).
 3. При $S = 7$ выигрышная стратегия есть у Вали. После первого хода Паши в куче может стать либо 8, либо 14 камней. В обеих этих позициях выигрывает игрок, который будет делать ход (теперь это Валя). Случай $S = 8$ рассмотрен в п. 2, случай $S = 14$ рассмотрен в п. 1а.
- В таблице изображено дерево возможных партий при описанной стратегии Вали. Заключительные позиции (в них выигрывает Валя) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов						
И.п.	1-й ход Паши (все ходы)	1-й ход Вали (только ход по стратегии)	2-й ход Паши (все ходы)	2-й ход Вали (только ход по стратегии)	3-й ход Паши (все ходы)	3-й ход Вали (только ход по стратегии)
7	$7+1=8$	$8*2=16$	$16+1=17$	$17+1=18$	$18+1=19$	$19+1=20$
			$16*2=32$		$18*2=36$	
	$7*2=14$	$14*2=28$				

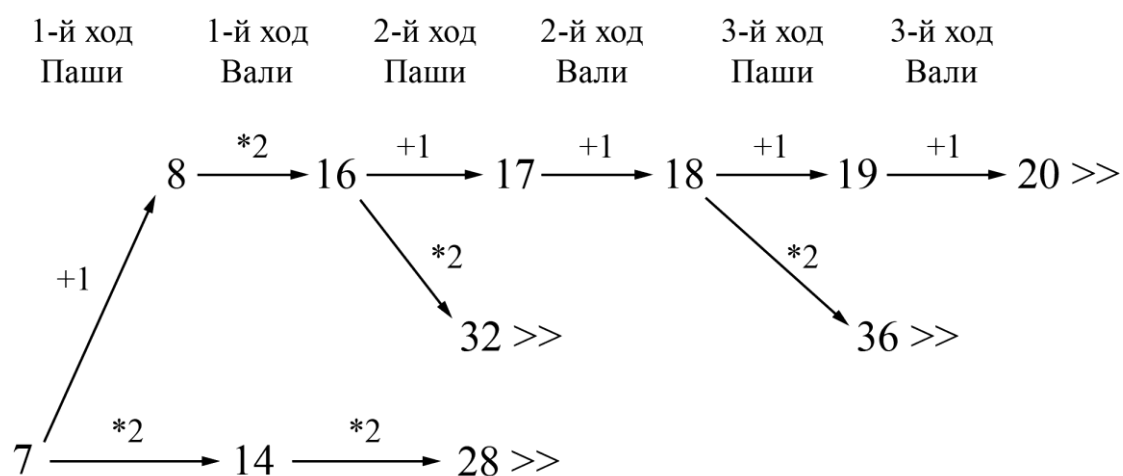


Рис.1. Дерево всех партий, возможных при Валиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

Указания по оцениванию	Баллы
<p>Предварительные замечания</p> <p>В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>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.</p> <p>Пункт 1а считается выполненным, если правильно указаны все позиции, в которых Паша выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт 1б считается выполненным, если (i) правильно указано, кто из игроков имеет выигрышную стратегию в каждой из указанных позиций, и (ii) описаны выигрышные стратегии – так, как это сделано в образце решения, или другим способом. Первое задание считается выполненным полностью, если выполнены полностью оба пункта: 1а и 1б.</p>	

Замечание для проверяющего. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника (см. условие задачи). Есть два основных способа сделать это. (1) Можно построить дерево всех партий, возможных при выбранной стратегии, и убедиться, что все заключительные позиции являются выигрышными для игрока, реализующего стратегию. (2) Можно свести задачу к рассмотренным выше позициям. Например, выигрышную стратегию для игрока, который ходит первым, можно описать, указав ход, который ведёт в позицию, для которой известна выигрышная стратегия для игрока, который ходит вторым. Чтобы подобным образом описать выигрышную стратегию для игрока, который ходит вторым (Вали), нужно перебрать все возможные первые ходы Паши и убедиться, что для всех полученных позиций мы знаем выигрышную стратегию для игрока, который ходит первым. В примере решения мы используем в основном второй способ описания стратегии. Экзаменуемый может описывать стратегию любым удобным ему способом. Существенно (повторим), чтобы (1) для каждой позиции, которая может встретиться игроку, реализующему стратегию, было понятно, какой ход он должен сделать, и (2) было показано, что все возможные заключительные позиции выигрышные для этого игрока.

Задание 2 считается выполненным, если (i) правильно указано, кто из игроков имеет выигрышную стратегию в каждой из указанных позиций, и (ii) описаны выигрышные стратегии.

Задание 3 считается выполненным, если (i) правильно указано, что выигрышную стратегию имеет Валя; (ii) правильно описано дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). При этом допускаются арифметические ошибки, не искажающие сути решения.

Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом

Выполнены второе и третье задания. Для первого задания правильно перечислены позиции, в которых Паша выигрывает первым ходом (п. 1а) и правильно указано, кто из игроков имеет выигрышную стратегию при указанных значениях S (п. 1б). При этом допускаются недочёты следующих типов: – в п. 1а не указано, каким ходом выигрывает Паша; – в п. 1б не указано, что игрокам нет смысла удваивать количество камней в куче. Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. <ul style="list-style-type: none">• Выполнено третье задание.• Выполнены первое и второе задания.• Первое задание выполнено, возможно, при наличии недочётов, указанных в критериях на 3 балла; для второго задания (i) правильно указано, кто из игроков имеет выигрышную стратегию в каждой из указанных позиций, и (ii) правильно указан первый ход Паши при выигрышной стратегии, однако не указано, что после выбранного хода Паши получается позиция, выигрышная для Вали; для третьего задания правильно указан игрок, имеющий выигрышную стратегию	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. <ul style="list-style-type: none">• Первое задание выполнено, возможно, с недочётами, указанными в критериях на 3 балла.• Второе задание выполнено, возможно, с недочётами, указанными в критериях на 2 балла.• Для второго и третьего заданий во всех случаях правильно указан игрок, имеющий выигрышную стратегию	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
Максимальный балл	3

27

*Вам предлагается два задания с похожими условиями: задание А и задание Б. Вы можете решать оба задания или одно из них по своему выбору. Задание Б более сложное, его решение оценивается выше. Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б.*

Задание А. Имеется набор данных, состоящий из 6 пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Напишите программу для решения этой задачи. В этом варианте задания оценивается только правильность программы, время работы и размер использованной памяти не имеют значения.

Максимальная оценка за правильную программу – 2 балла.

Задание Б. Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Напишите программу для решения этой задачи.

Постарайтесь сделать программу эффективной по времени и по используемой памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству пар чисел N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но не эффективную по памяти, – 3 балла.

Как в варианте А, так и в варианте Б программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи (или 0, если такую сумму получить нельзя).

НАПОМИНАЕМ! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Перед текстом программы кратко опишите Ваш алгоритм решения, укажите использованный язык программирования и его версию (например, Free Pascal 2.6.4).

Входные данные

Для варианта А на вход программе подаётся 6 строк, каждая из которых содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта А:

```
1 3
5 12
6 9
5 4
3 3
1 1
```

Для варианта Б на вход программе в первой строке подаётся количество пар N ($1 \leq N \leq 100\,000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта Б:

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

Пример выходных данных для приведённых выше примеров входных данных:

32

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

Задание Б. Сначала рассмотрим решение для более общего задания (вариант Б).

Решение 1.

Чтобы получить максимально возможную сумму, будем брать из каждой пары самое большое число. Если полученная при этом сумма будет делиться на 3, её необходимо уменьшить. Для этого достаточно в одной из пар, где числа имеют разные остатки при делении на 3, заменить ранее выбранное число на другое число из той же пары. При этом разница между числами в паре должна быть минимально возможной. Если во всех парах оба числа имеют одинаковый остаток при делении на 3, получить нужную сумму невозможно.

Замечание для эксперта. От ученика не требуется доказывать правильность предложенного алгоритма. Для удобства экспертов докажем, что при наличии решения достаточно заменить одно число. Пусть это не так, т.е. найдутся две такие пары, от которых в искомую сумму входят не бóльшие в своих парах числа x_1 и y_1 , а меньшие числа из соответствующих пар: x_2 и y_2 . При этом $x_2 + y_2$ имеет остаток от деления на 3, отличный от остатка от деления на 3 числа $x_1 + y_1$ (иначе мы могли бы включить в сумму $x_1 + y_1$ вместо $x_2 + y_2$). Но это означает, что хотя бы одно из чисел x_2, y_2 тоже при делении на 3 имеет остаток, отличный от соответствующего максимального числа пары. Значит, оптимальной является замена только одного из таких чисел.

Программа читает все данные один раз. В каждой паре определяется большее число Max и разность между бóльшим и меньшим числами пары D . После обработки очередной пары программа хранит два числа: s – сумму всех максимальных элементов прочитанных пар и D_{min} – наименьшую возможную разность D , не кратную 3. Окончательным ответом будет значение s , если оно не делится на 3, и $s - D_{min}$ в противном случае. Если s делится на 3, а D_{min} не определено (разность между числами во всех парах кратна 3), ответ в соответствии с условиями задачи считается равным 0

Программа 1. Пример правильной и эффективной программы для задания Б на языке Паскаль

```
const
    aMax = 10000; {наибольшее возможное число в исходных данных}

var
    N: longint; {количество пар}
    a, b: longint; {пара чисел}
    Max: longint; {максимум в паре}
    Min: longint; {минимум в паре}
    s: longint; {сумма выбранных чисел}
    D_min: longint; {минимальная разница Max-Min не кратная 3}
    i: longint;

begin
    s := 0;
    D_min := aMax + 1;
    readln(N);
    for i := 1 to N do begin
        readln(a, b);
        if a>b then begin Max:=a; Min:=b end
            else begin Max:=b; Min:=a end;
        s := s + Max;
        if ((Max - Min) mod 3 > 0) and (Max - Min < D_min)
            then D_min := Max - Min
        end;
        if s mod 3 = 0 then begin
            if D_min > aMax then s := 0
            else s := s - D_min
        end;
        writeln(s)
    end.
end.
```

Решение 2.

Возможно и решение, основанное на другой идее. А именно, будем хранить для каждого прочитанного набора пар три суммы (s_0 , s_1 , s_2) – максимальные суммы элементов пар, имеющие при делении на 3 соответственно остатки 0, 1 и 2. При обработке очередной пары (a_1 , a_2) эти суммы обновляются. Для этого достаточно рассмотреть суммы s_0+a_1 , s_1+a_1 , s_2+a_1 , s_0+a_2 , s_1+a_2 , s_2+a_2 и для каждого возможного остатка от деления на 3 выбрать в качестве нового значения s_0 , s_1 или s_2 значение наибольшей из указанных сумм, дающей данный остаток. Окончательным ответом будет бóльшая из сумм s_1 и s_2 .

Эта идея приводит к более громоздкой реализации, но все основные требования по эффективности в ней выполнены, поэтому подобное решение при отсутствии ошибок можно оценить максимальным количеством баллов.

Ниже приводится пример основанной на этом принципе программы на языке

Паскаль.

Замечание для эксперта. В приведённом ниже решении для хранения s_0 , s_1 , s_2 используется массив $s_new[0..2]$. Это упрощает реализацию, однако решение, в котором используются простые переменные, также допустимо. Не следует снижать баллы только за то, что в программе использованы простые переменные, а не массив, как в приведённом ниже примере

Программа 2. Пример правильной и эффективной программы для задания Б на языке Паскаль

```
var
  N: longint; {количество пар}
  a: array[1..2] of longint; {пара чисел}
  s_old, s_new: array[0..2] of longint;
    {суммы с соответствующими остатками от деления на 3}
  i, j, k, r: longint;
begin
  readln(N);
  for j := 0 to 2 do
    s_old[j] := 0;
  for i := 1 to N do begin
    readln(a[1], a[2]);
    for j := 0 to 2 do
      s_new[j] := 0;
    for k := 1 to 2 do begin
      for j := 0 to 2 do begin
        if (s_old[j] > 0) or (i = 1) then begin
          r := (s_old[j] + a[k]) mod 3;
          if s_new[r] < s_old[j] + a[k] then
            s_new[r] := s_old[j] + a[k]
        end
      end
    end;
    s_old := s_new
  end;
  if s_new[1] > s_new[2] then
    writeln(s_new[1])
  else
    writeln(s_new[2]);
  {если решения не существует, то s_new[1] и s_new[2]
   окажутся равными нулю}
end.
```

Замечание для эксперта. Ученик может «перестраховаться» и явно проверить, что хотя бы одно из чисел `s_new[1]`, `s_new[2]` отлично от 0. Эта проверка излишня (см. комментарий в конце программы), однако она не влияет на порядок роста времени программы. Снижать баллы за такую избыточную проверку не следует.

Задание А. Это задание можно выполнить «в лоб»: сохранить в массиве все исходные данные, перебрать все возможные способы выбора одного элемента из каждой пары и найти максимальную сумму, соответствующую условиям задачи.

Ниже приводится пример такого решения

Пример решения задачи А на языке Паскаль

```
var
  a: array[1..6, 1..2] of longint;
  i1, i2, i3, i4, i5, i6: longint;
  s, sMax: longint;
begin
  for i1:= 1 to 6 do readln(a[i1,1], a[i1,2]);
  sMax := 0;
  for i1:=1 to 2 do
    for i2:=1 to 2 do
      for i3:=1 to 2 do
        for i4:=1 to 2 do
          for i5:=1 to 2 do
            for i6:=1 to 2 do begin
              s:=a[1,i1]+a[2,i2]+a[3,i3]+a[4,i4]+a[5,i5]+a[6,i6];
              if (s mod 3 <> 0) and (s > sMax) then sMax := s
            end;
          writeln(sMax)
        end.
      end.
    end.
  end.
```

Указания по оцениванию	Баллы
<p><i>Предварительные замечания.</i></p> <p>1. В задаче есть два задания (А и Б). Соответственно, ученик может представить две программы. В каждой из программ должно быть указано, решением какого из заданий она является. Если в работе представлена одна программа, то в ней также должно быть указано, решением какого из заданий она является.</p> <p>2. Если ученик не указал, к какому заданию относится программа, или можно предположить, что ученик ошибся в идентификации программ, необходимо следовать приведённым ниже инструкциям.</p> <p>Случай 2.1. Ученик представил только одну программу. Следует рассматривать программу как решение задания Б и оценивать её по соответствующим критериям.</p> <p>Случай 2.2. Ученик представил две программы, но указание задания есть только для одной из программ. Следует рассматривать вторую программу как ответ на оставшееся задание.</p> <p>Случай 2.3. Ученик представил две программы; ни для одной из них задание не указано, или в обоих решениях указано одно и то же задание. Следует первую (по порядку в представленных учеником материалах) программу рассматривать как ответ на задание А, а вторую – как ответ на задание Б.</p> <p>Случай 2.4. Ученик представил более двух программ. Следует рассматривать только две последние программы и соотносить их с заданиями по правилам 2.1–2.3.</p> <p>Случай 2.5. Решение, представленное в качестве решения задания А, по критериям для задания Б может быть оценено в 3 или 4 балла. При этом решение, представленное в качестве решения задания Б, оценено меньшим баллом. Следует считать, что ученик перепутал обозначения заданий и оценивать решение, представленное как решение задания А, по критериям задания Б.</p> <p>НАПОМИНАЕМ! <i>Итоговый балл за задачу – это бóльший из баллов, полученных учеником за каждое из двух представленных решений.</i></p>	

Пояснения для проверяющих.

1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то считается, что учеником допущена опечатка, и решение оценивается по критериям для задания А.

В качестве решения задания А может быть представлена программа, которая решает задачу при произвольном количестве входных данных (как в задании Б), осуществляя полный перебор всех вариантов. Такая программа неэффективна по времени, поэтому она должна оцениваться 2 баллами.

2. Два задания (и, соответственно, возможность для экзаменуемого представить две программы) дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое даёт ему право получить 2 балла, а затем приступить к поиску более эффективного решения.

3. Приведённые в п. 2.1–2.5 правила имеют целью избежать снижения баллов из-за того, что ученик перепутал обозначения заданий.

Общие принципы оценивания решений

4 балла ставится за эффективную и правильно работающую программу, решающую задачу в общем случае (задача Б). При этом программа может содержать до трёх синтаксических ошибок («описок»).

3 балла ставится в случае, когда фактически задача решена и решена эффективно по времени, возможно, с хранением всех входных данных в массиве, но количество «описок» более трёх (но не более пяти) и допущено не более одной содержательной ошибки, не позволяющей усомниться в том, что экзаменуемый правильно придумал алгоритм.

2 балла ставится, если программа, решающая задачу Б, в дополнение к неточностям, которые перечислены выше, работает неэффективно по времени и/или допущено до трёх упомянутых выше содержательных ошибок. Количество допустимых «описок» – до семи.

2 балла также ставится за правильное решение упрощенной задачи (задача А).

1 балл ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.

Далее сказанное уточнено

Критерии оценивания задания А	
При решении задачи А программа верно находит требуемую сумму для любых 6 пар исходных данных. Допускается до пяти синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла)	2
Не выполнены условия, позволяющие поставить 2 балла. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. Допускается любое количество «описок»	1
Не выполнены критерии, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл для задания А</i>	2
Критерии оценивания задания Б	
Программа правильно работает для любых соответствующих условию входных данных и при этом эффективна как по времени, так и по памяти, т.е. не используются массивы и другие структуры данных (в том числе стек рекурсивных вызовов), размер которых зависит от количества входных элементов, а время работы пропорционально этому количеству. Возможно использование массивов и динамических структур данных при условии, что в них в каждый момент времени хранится фиксированное количество элементов, требующих для хранения меньше 1Кб. Программа может содержать не более трёх синтаксических ошибок следующих видов: 1) пропущен или неверно указан знак пунктуации; 2) неверно написано или пропущено зарезервированное слово языка программирования; 3) не описана или неверно описана переменная; 4) применяется операция, недопустимая для соответствующего типа данных. К синтаксическим ошибкам приравнивается использование неверного типа данных. Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку	4

<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел, правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел. Количество синтаксических ошибок («описок») указанных выше видов – не более пяти.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в C++ или другой структуре данных).</p> <p>Допускается ошибка при вводе и выводе данных, не влияющая на содержание решения.</p> <p>Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла.</p> <p>Кроме того, допускается наличие одной ошибки, принадлежащей к одному из следующих видов:</p> <ol style="list-style-type: none"> 1) ошибка инициализации, в том числе отсутствие инициализации; 2) не выводится результат, равный 0, или вместо 0 выводится неверное значение; 3) допущен выход за границу массива; 4) используется знак “<” вместо “<=”, “or” вместо “and” и т.п. 	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла.</p> <p>Программа работает в целом верно, эффективно или нет, например для решения задачи используется перебор всех возможных вариантов выбора элементов в парах. В реализации алгоритма допускается до трёх содержательных ошибок, допустимые виды ошибок перечислены в критериях на 3 балла.</p> <p>Количество синтаксических «описок» не должно быть более семи.</p> <p>Программа может быть неэффективна по времени, например все числа запоминаются в массиве и перебираются все возможные суммы, т.е., по сути, реализовано решение задачи А без ограничений на количество введённых пар</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.</p> <p>Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. Допускается любое количество «описок». 1 балл ставится также за решения, верные лишь в частных случаях</p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
Максимальный балл для задания Б	4
Итоговый максимальный балл	4