

### Критерии оценивания заданий с развёрнутым ответом

- 24** Дано целое положительное число  $N$ . Необходимо найти наибольшее количество подряд идущих единиц в десятичной записи  $N$ . Например, для  $N = 2019$  нужно получить ответ 1, а для  $N = 21011$  ответ 2. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, M, K AS INTEGER INPUT N M = 0 K = 0 WHILE N &gt;= 10     IF N MOD 10 = 1 THEN         K = K + 1     ELSE         K = 1     END IF     IF K &gt; M THEN         M = M + 1     END IF     N = N \ 10 WEND PRINT M </pre>	<pre> n = int(input()) m = 0 k = 0 while n &gt;= 10:     if n % 10 == 1:         k += 1     else:         k = 1     if k &gt; m:         m += 1     n //= 10 print(m) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел n, m, k     ввод n     m := 0     k := 0     нц пока n &gt;= 10         если mod(n,10) = 1             то k := k + 1             иначе k := 1         все         если k &gt; m             то m := m + 1         все         n := div(n,10)     кц     вывод m кон </pre>	<pre> var n, m, k: integer; begin     readln(n);     m := 0;     k := 0;     while n &gt;= 10 do begin         if n mod 10 = 1             then k := k + 1             else k := 1;         if k &gt; m then             m := m + 1;         n := n div 10;     end;     writeln(m) end. </pre>

**C++**

```

#include <iostream>
using namespace std;
int main() {
    int n, m, k;
    cin >> n;
    m = 0;
    k = 0;
    while (n >= 10) {
        if (n % 10 == 1)
            k += 1;
        else
            k = 1;
        if (k > m)
            m += 1;
        n /= 10;
    }
    cout << m;
    return 0;
}

```

Последовательно выполните следующее.

1. Приведите пример числа  $N$ , при котором программа выведет верный ответ. Укажите этот ответ.
2. Приведите пример числа  $N$ , при котором программа выведет неверный ответ. Укажите верный ответ и ответ программы.
3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. 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
(допускаются иные формулировки ответа, не искажающие его смысла)
<ol style="list-style-type: none"> <li>1. Примеры чисел, при вводе которых программа выводит верные ответы: 15, (ответ 1), 51 (ответ 1), 711 (ответ 2), 2111 (ответ 3).</li> <li>2. Примеры чисел, при вводе которых программа выводит неверные ответы: 2121 (правильный ответ 1, программа выводит 2), 512 (правильный ответ 1, программа выводит 2), 111 (правильный ответ 3, программа выводит 2).</li> <li>3. Программа содержит две ошибки.</li> </ol>
<p><b>Первая ошибка.</b> Неверное условие цикла. Из-за этой ошибки первая цифра числа не проверяется</p>

**Вторая ошибка.** Неверный сброс счётчика единиц  $k$ . Из-за этой ошибки длина цепочки единиц, расположенных не в конце числа, оказывается на единицу больше верной.

**Не является ошибкой** увеличение  $m$  на 1. Хотя более логичным кажется присваивание  $m := k$ , способ, приведённый в программе, тоже работает, так как  $k$  всегда оказывается больше  $m$  ровно на 1.

### Пример исправления для алгоритмического языка

#### Первая ошибка:

нц пока  $n \geq 10$

Исправленная строка, способ 1:

нц пока  $n \geq 1$

Исправленная строка, способ 2:

нц пока  $n > 0$

#### Вторая ошибка:

иначе  $k := 1$

Исправленная строка:

иначе  $k := 0$

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

Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков, не влияющих на содержательную часть исправления (например, слова «иначе» или точки с запятой).

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать пример входного числа, при котором программа выдаёт верный ответ, и ответ программы в этом случае.</li> <li>2. Указать пример входного числа, при котором программа выдаёт неверный ответ, верный ответ и ответ программы в этом случае.</li> <li>3. Исправить первую ошибку в программе.</li> <li>4. Исправить вторую ошибку в программе.</li> </ol> <p>Действие 1 считается выполненным, если указаны входное значение и ответ; при выполнении программы с этим входным значением получается этот ответ, и этот ответ совпадает с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Действие 2 считается выполненным, если указаны входное значение, верный ответ и ответ программы; при выполнении программы с этим входным значением получается указанный ответ программы, и указанный верный ответ совпадает с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Для действий 1 и 2 экзаменуемый не обязан объяснять, как получен результат, достаточно указать верные числа</p>	

Каждое из действий 3 и 4 считается выполненным при одновременном выполнении двух условий: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка)	
Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. 1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной. 2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Дан массив, содержащий 2019 положительных целых чисел, не превышающих 15 000. Необходимо найти минимальный и максимальный **нечётные** элементы массива (если в массиве нет нечётных элементов, минимум и максимум считаются равными нулю), вычислить их среднее арифметическое, уменьшить все **чётные** элементы, превышающие это среднее, на величину этого среднего и вывести изменённый массив. Например, для исходного массива из пяти элементов 30, 89, 27, 90, 68 программа должна вывести числа 30, 89, 27, 42, 16 (минимум и максимум равны 27 и 89, их среднее равно 58, все чётные элементы, превышающие 58, уменьшены на 58).

Напишите на одном из языков программирования программу для решения этой задачи.

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

Бейсик	Python
<pre> CONST N=2019 DIM A(N) AS INTEGER DIM I, M, K AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END </pre>	<pre> # кроме уже указанных # допускается использование # целочисленных переменных # m, k a = [] N = 2019 for i in range(0, N):     a.append(int(input())) ... </pre>

Алгоритмический язык	Паскаль
алг нач цел N=2019 целтаб a[1:N] цел i, m, k нц для i от 1 до N ввод a[i] кц ... Кон	const N=2019; var a: array [1..N] of integer; i, m, k: integer; begin for i:=1 to N do readln(a[i]); ... end.
C++	
<pre>#include &lt;iostream&gt; using namespace std; const int N=2019; int main(){     int a[N];     int i, m, k;     for (i=0; i&lt;N; ++i)         cin &gt;&gt; a[i];      ...     return 0; }</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 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
Задача решается в два прохода: на первом проходе определяются минимум и максимум, после этого вычисляется их среднее, и на втором проходе производится корректировка и вывод элементов. Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод
Пример правильной программы на языке Паскаль
<pre>m := 15001; k := 0; for i:=1 to N do begin     if (a[i] mod 2 = 1) and (a[i] &lt; m) then m := a[i];     if (a[i] mod 2 = 1) and (a[i] &gt; k) then k := a[i]; end; if m &gt; 15000 then m := 0; m := (m+k) div 2; for i:=1 to N do begin     if (a[i] mod 2 = 0) and (a[i] &gt; m) then         a[i] := a[i] - m;</pre>

<pre>writeln(a[i]) end;</pre>	
<p>При использовании языка Python для нахождения минимума и максимума можно применить функции <code>min</code> и <code>max</code>. При этом обязательно нужно использовать параметр <code>default</code> (доступен начиная с версии Python 3.4) или другим способом обеспечить обработку ситуации, когда в массиве нет нечётных элементов</p>	
<p><b>Пример правильной программы на языке Python</b></p> <pre>m = min((i for i in a if i%2 == 1), default = 0) k = max((i for i in a if i%2 == 1), default = 0) m = (m+k) // 2 for i in range(0,N):     if a[i]%2 == 0 and a[i] &gt; m:         a[i] -= m     print(a[i])</pre>	
<p>Использовать описанную возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на языке Паскаль</p>	
Указания по оцениванию	Баллы
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</p> <p>Не считается ошибкой и не приводит к снижению оценки использование для определения среднего значения обычной операции деления вместо деления нацело (в тех языках, где эти операции различаются).</p> <p>Эффективность не имеет значения и не оценивается.</p> <p>Допускается запись программы на языке, не входящем в список языков из условия. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если выбранный 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.</p> <p>Допускается изменение указанного в условии формата вывода, например, вывод всех элементов массива в одну строку</p>	
Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив	2
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку)</p> <p>1) Отсутствие инициализации или неверная инициализация</p>	1

минимума или максимума. 2) Неверное определение минимума или максимума. 3) Неверное определение среднего значения. 4) Неверное построение логических условий (поиск минимума или максимума на неверном множестве). 5) Выход за границы массива. 6) Исходный массив не изменяется. 7) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться. 8) Не учитывается или неверно обрабатывается ситуация, когда в исходном массиве нет нечётных элементов. 9) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы). 10) Используется переменная, не объявленная в разделе описания переменных. 11) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно	
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
Максимальный балл	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один камень**, **увеличить** количество камней **в первой куче в два раза** или **увеличить** количество камней **во второй куче в три раза**. Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (12, 9), (6, 10), (6, 27). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 84. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 84 или больше камней.

В начальный момент в первой куче было 16 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 67$ .

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. В описание выигрышной стратегии не следует включать ходы играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

**Задание 1.**

Назовите все значения  $S$ , при которых Петя может выиграть первым ходом.

**Задание 2.**

Укажите такое значение  $S$ , при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанного значения  $S$  опишите выигрышную стратегию Пети.

**Задание 3.**

Укажите такое значение  $S$ , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом 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). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

Содержание верного ответа
---------------------------

(допускаются иные формулировки ответа, не искажающие его смысла)
--

**Задание 1.**

Петя может выиграть первым ходом, если  $S = 23, \dots, 67$ . Для выигрыша достаточно утроить количество камней во второй куче. При меньших значениях  $S$  за один ход нельзя получить 84 или более камней в двух кучах.

**Задание 2.**

Возможные значения  $S$ : 17, 22. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при  $S = 17$  Петя может получить позицию (32, 17), а при  $S = 22$  – позицию (17, 22).

В первом случае после хода Вани возникнет одна из позиций (33, 17), (64, 17), (32, 18), (32, 51), во втором случае – одна из позиций (18, 22), (34, 22), (17, 23), (17, 66). В любой из перечисленных позиций Петя может выиграть, утроив количество камней во второй куче.

**Задание 3.**

Возможное значение  $S$ : 21. После первого хода Пети возможны позиции (17, 21), (32, 21), (16, 22), (16, 63). В позициях (32, 21) и (16, 63) Ваня может выиграть первым ходом, утроив количество камней во второй куче. Из позиций (17, 21) и (16, 22) Ваня может получить позицию (17, 22), разобранный в задании 2. Игрок, после хода которого возникла эта позиция (в данном случае – Ваня), выигрывает следующим ходом.

В таблице изображены возможные партии при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом. На рисунке эти же партии показаны в виде графа (оба способа изображения допустимы)



	Положения после очередных ходов			
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(16, 21) Всего 37	(32,21) Всего 55	(32, 63) <b>Всего 95</b>		
	(16, 63) Всего 79	(16, 189) <b>Всего 205</b>		
	(17, 21) Всего 38	(17, 22) Всего 39	(18, 22) Всего 40	(18, 66) <b>Всего 84</b>
	(16, 22) Всего 38		(34, 22) Всего 56	(34, 66) <b>Всего 100</b>
			(17, 23) Всего 40	(17, 69) <b>Всего 86</b>
			(17, 66) Всего 83	(17, 198) <b>Всего 215</b>

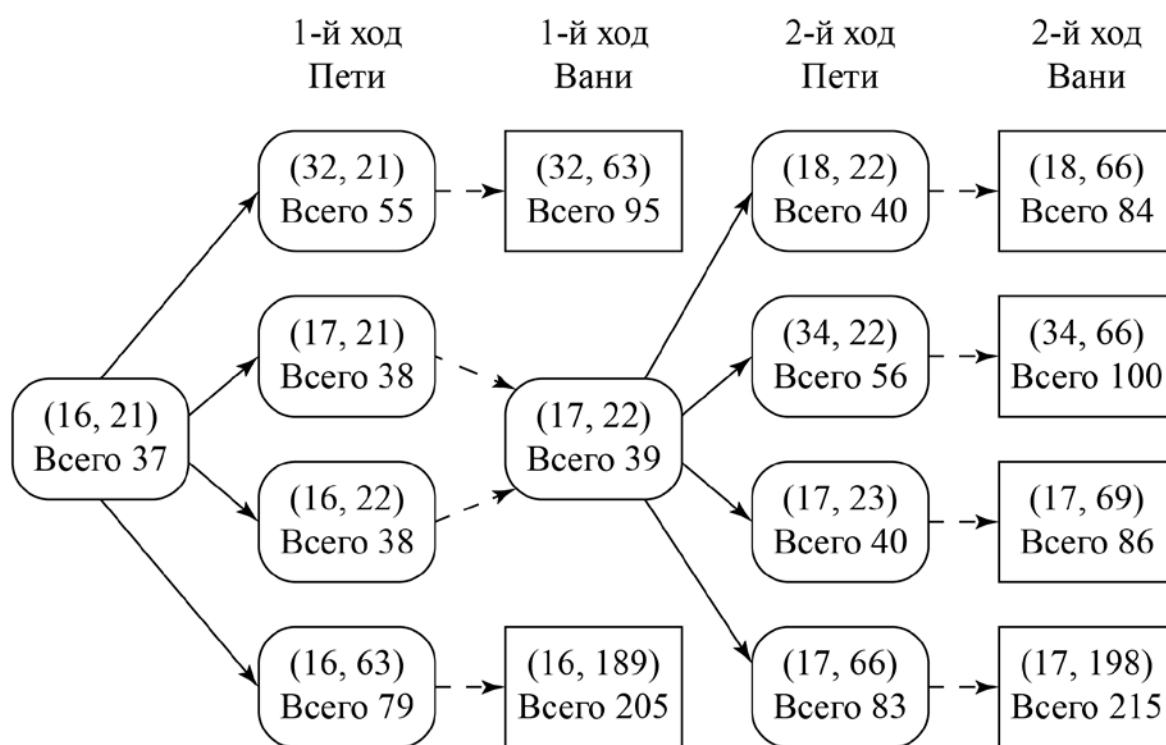


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани показаны пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*Примечание для эксперта.* Дерево всех партий может быть изображено в виде таблицы или в виде ориентированного графа – так, как показано на рисунке,

или другим способом. Например, вместо приведённого здесь «экономного» варианта, в котором позиции не дублируются, возможно построение полного дерева, в котором одинаковые позиции, возникающие при различном ходе игры, показаны отдельно. Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии со множеством партий, возможных при описанной в решении стратегии. В некоторых позициях заключительный выигрывающий ход можно сделать несколькими способами. В таблице и на рисунке указан один из них, в работе допускается выбор любого допустимого заключительного выигрывающего хода.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить <b>три</b> задания. 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>Задание 1 выполнено, если перечислены все удовлетворяющие условию значения <math>S</math>, и только они. Обоснование найденных значений не обязательно.</p> <p>Задание 2 выполнено, если верно указана выигрышная для Пети позиция (любая из двух возможных) и описана соответствующая стратегия.</p> <p>Задание 3 выполнено, если правильно указана выигрышная для Вани позиция и построено дерево всех возможных при выигрышной стратегии партий (и только их).</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом</p>	
Выполнены все три задания	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий.</p> <ul style="list-style-type: none"> <li>– Выполнено задание 3.</li> <li>– Выполнены задания 1 и 2</li> </ul>	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из заданий 1 и 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

27

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, находящихся на расстоянии не меньше 6 (разница в индексах элементов должна быть 6 или более). Необходимо определить количество пар, сумма чисел в которых кратна 3.

**Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $6 \leq N \leq 1000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

8  
1  
3  
5  
4  
6  
7  
9  
8

*Пример выходных данных для приведённого выше примера входных данных:*

1

*Пояснение.* Из восьми чисел можно составить три пары, удовлетворяющие условию. Это будут элементы с индексами 1 и 7, 1 и 8, 2 и 8. Для заданного набора чисел получаем пары (1, 9), (1, 8), (3, 8). Суммы чисел в этих парах равны 10, 9, 11. Одна из этих сумм кратна 3.

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает одного килобайта и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **бóльшая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

### Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Сумма двух чисел кратна 3, если сумма остатков от деления этих чисел на 3 равна 3.

Будем хранить количество чисел, дающих при делении на 3 остатки 0, 1 и 2 без учёта шести последних элементов последовательности. Для этого понадобится хранить последние шесть элементов. Остальные элементы последовательности можно не хранить, это обеспечивает эффективность по памяти. Для хранения шести элементов можно использовать циклический массив, как показано в решении 1.

Будем рассматривать каждое введенное число как правый элемент возможной пары (первые шесть чисел не могут быть таким элементом) и в зависимости от остатка при его делении на 3 определять количество подходящих пар и прибавлять его к общей сумме

### Решение 1. Правильная и эффективная программа на языке Паскаль (циклический массив, счётчики в отдельном массиве)

```
const d=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[0..d-1] of integer;
  k: array[0..2] of integer; {количество по остаткам}
  s: integer;      {количество подходящих пар}
  i: integer;      {счётчик для ввода}
  ia: integer;     {текущий индекс в массиве a}
begin
  readln(N);
  {ввод первых d чисел}
  for i:=0 to d-1 do readln(a[i]);
  {ввод и обработка остальных значений}
  for i:=0 to 2 do k[i]:=0;
  ia:=0; s:=0;
  for i:=d to N-1 do begin
    readln(x);
    k[a[ia] mod 3] := k[a[ia] mod 3] + 1;
    s := s + k[(3 - x mod 3) mod 3];
    a[ia] := x;
    ia := (ia+1) mod d
  end;
  writeln(s)
end.
```

Вместо циклического массива можно использовать сдвиги. В этом случае для вычисления всегда используется первый элемент массива, а новое число

записывается в последний. Хотя этот алгоритм работает медленнее, чем алгоритм с циклическим массивом (для каждого элемента требуется пять дополнительных присваиваний при сдвигах), основное требование эффективности здесь выполнено: при увеличении размера массива в  $k$  раз количество действий растёт не более чем в  $k$  раз.

Для хранения счётчиков по остаткам можно вместо массива из трёх элементов использовать три отдельные переменные. Это делает программу чуть более громоздкой, но не влияет на эффективность.

Ниже приводится пример программы, в которой использованы эти методы

## **Решение 2. Правильная и эффективная программа на языке Паскаль (сдвиг массива, счётчики в отдельных переменных)**

```
const d=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[1..d] of integer;
  k0, k1, k2: integer; {количество по остаткам}
  s: integer;      {количество подходящих пар}
  i: integer;      {счётчик для ввода}
  ia: integer;     {счётчик для сдвига}
begin
  readln(N);
  {ввод первых d чисел}
  for i:=1 to d do readln(a[i]);
  {ввод и обработка остальных значений}
  k0 := 0; k1 := 0; k2:=0;
  s:=0;
  for i:=d+1 to N do begin
    readln(x);
    if a[1] mod 3 = 0 then k0 := k0 + 1;
    if a[1] mod 3 = 1 then k1 := k1 + 1;
    if a[1] mod 3 = 2 then k2 := k2 + 1;
    if x mod 3 = 0 then s := s + k0;
    if x mod 3 = 1 then s := s + k2;
    if x mod 3 = 2 then s := s + k1;
    for ia:=1 to s-1 do a[ia]:=a[ia+1];
    a[s] := x;
  end;
  writeln(s)
end.
```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и подсчитаем количество подходящих. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растёт квадратично). Такая программа оценивается не выше 2 баллов

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)

**Решение 3. Правильная, но неэффективная программа на языке Паскаль**

```

const d=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  a: array [1..1000] of integer; {исходные данные}
  s: integer;      {количество подходящих пар}
  i,j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  s :=0;
  for i := 1 to N-d do begin
    for j := i+d to N do begin
      if (a[i] + a[j]) mod 3 = 0
      then s := s +1
    end
  end;
  writeln(s)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается в 0 баллов	
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> <li>1) пропущен или неверно указан знак пунктуации;</li> <li>2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</p>	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.</p>	3

<p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в C++ или другой аналогичной структуре данных).</p> <p>Количество синтаксических ошибок (описок), указанных в критериях на 4 балла, – не более пяти.</p> <p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (не считывается значение <math>N</math> или неверно организован ввод последовательности);</li> <li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li> <li>3) используется неверный тип данных;</li> <li>4) использована одна переменная (константа) вместо другой;</li> <li>5) используется один знак операции вместо другого;</li> <li>6) отсутствует вывод ответа или выводится не то значение (хотя правильный ответ в программе найден);</li> <li>7) неверная работа с массивом, в том числе выход за границы массива;</li> <li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками)</li> </ol>	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.</p> <p>ИЛИ Программа работает в целом верно и эффективно по времени, но в ней есть не подходящие под перечень из критериев на 3 балла ошибки, которые в некоторых особых случаях приводят к неверным результатам.</p> <p>ИЛИ Представлено корректное переборное решение, в котором все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например, выход индексов за границы массива, неверный учёт расстояния между элементами и т. д.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа представлена и содержит как минимум два обязательных элемента, возможно, реализованных с ошибками:</p> <ol style="list-style-type: none"> <li>1) рассматриваются только пары, находящиеся на расстоянии не меньше заданного в условии;</li> <li>2) ведётся подсчёт пар с подходящей суммой</li> </ol>	1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл</i>	4

### Критерии оценивания заданий с развёрнутым ответом

24

Дано целое положительное число  $N$ . Необходимо найти наибольшее количество подряд идущих нулей в десятичной записи  $N$ . Например, для  $N = 2019$  нужно получить ответ 1, а для  $N = 20\ 100$  ответ 2.

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, M, K AS INTEGER INPUT N M = 0 K = 0 WHILE N &gt;= 10     IF N MOD 10 = 0 THEN         K = K + 1     ELSE         K = 1     END IF     IF K &gt; 0 THEN         M = K     END IF     N = N \ 10 WEND PRINT M </pre>	<pre> n = int(input()) m = 0 k = 0 while n &gt;= 10:     if n % 10 == 0:         k += 1     else:         k = 1     if k &gt; 0:         m = k     n //= 10 print(m) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел n, m, k     ввод n     m := 0     k := 0     нц пока n &gt;= 10         если mod(n,10) = 0             то k := k + 1             иначе k := 1         все         если k &gt; 0             то m := k         все         n := div(n,10)     кц     вывод m кон </pre>	<pre> var n, m, k: integer; begin     readln(n);     m := 0;     k := 0;     while n &gt;= 10 do begin         if n mod 10 = 0             then k := k + 1             else k := 1;         if k &gt; 0 then             m := k;         n := n div 10;     end;     writeln(m) end. </pre>



**C++**

```

#include <iostream>
using namespace std;
int main() {
    int n, m, k;
    cin >> n;
    m = 0;
    k = 0;
    while (n >= 10) {
        if (n % 10 == 0)
            k += 1;
        else
            k = 1;
        if (k > 0)
            m = k;
        n /= 10;
    }
    cout << m;
    return 0;
}

```

Последовательно выполните следующее.

1. Приведите пример числа  $N$ , при котором программа выведет верный ответ. Укажите этот ответ.
2. Приведите пример числа  $N$ , при котором программа выведет неверный ответ. Укажите верный ответ и ответ программы.
3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. 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.

<b>Содержание верного ответа</b>
(допускаются иные формулировки ответа, не искажающие его смысла)
<ol style="list-style-type: none"> <li>1. Примеры чисел, при вводе которых программа выводит верные ответы: 50 (ответ 1), 700 (ответ 2), 1000 (ответ 3).</li> <li>2. Примеры чисел, при вводе которых программа выводит неверные ответы: 51 (правильный ответ 0, программа выводит 1), 701 (правильный ответ 1, программа выводит 2), 101 000 (правильный ответ 3, программа выводит 2).</li> <li>3. Программа содержит две ошибки.</li> </ol>
<p><b>Первая ошибка.</b> Неверный сброс счётчика нулей <math>k</math>. Из-за этой ошибки длина цепочки нулей, расположенных не в конце числа, оказывается на единицу больше верной</p>

**Вторая ошибка.** Неверное условие нахождения более длинной цепочки. Из-за этой ошибки программа выводит длину не наибольшей, а последней найденной (самой левой) цепочки нулей.

**Не является ошибкой** условие цикла. Проверка  $n \geq 10$  вместо  $n \geq 1$  приводит к тому, что первая цифра числа не проверяется, но, поскольку ноль не может быть первой цифрой, это не влияет на правильность ответа.

### Пример исправления для алгоритмического языка

#### Первая ошибка:

```
иначе k := 1
```

Исправленная строка:

```
иначе k := 0
```

#### Вторая ошибка:

```
если k > 0
```

Исправленная строка:

```
если k > m
```

Допускается также нестрогое сравнение:

```
если k >= m
```

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

Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков, не влияющих на содержательную часть исправления (например, слова «иначе» или точки с запятой)

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать пример входного числа, при котором программа выдаёт верный ответ, и ответ программы в этом случае.</li> <li>2. Указать пример входного числа, при котором программа выдаёт неверный ответ, верный ответ и ответ программы в этом случае.</li> <li>3. Исправить первую ошибку в программе.</li> <li>4. Исправить вторую ошибку в программе.</li> </ol> <p>Действие 1 считается выполненным, если указаны входное значение и ответ; при выполнении программы с этим входным значением получается этот ответ, и этот ответ совпадает с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Действие 2 считается выполненным, если указаны входное значение, верный ответ и ответ программы; при выполнении программы с этим входным значением получается указанный ответ программы, и указанный верный ответ совпадает с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Для действий 1 и 2 экзаменуемый не обязан объяснять, как получен результат, достаточно указать верные числа</p> <p>Каждое из действий 3 и 4 считается выполненным при одновременном выполнении двух условий:</p>	

а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка)	
Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. 1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной. 2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Дан массив, содержащий 2019 положительных целых чисел, не превышающих 15 000. Необходимо найти минимальный и максимальный **чётные** элементы (если в массиве нет чётных элементов, минимум и максимум считаются равными нулю), вычислить их среднее арифметическое, уменьшить все **нечётные** элементы, превышающие это среднее, на величину этого среднего и вывести изменённый массив. Например, для исходного массива из пяти элементов 30, 89, 27, 90, 68 программа должна вывести числа 30, 29, 27, 90, 68 (минимум и максимум равны 30 и 90, их среднее равно 60, все нечётные элементы, превышающие 60, уменьшены на 60).

Напишите на одном из языков программирования программу для решения этой задачи.

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

Бейсик	Python
<pre>CONST N=2019 DIM A(N) AS INTEGER DIM I, M, K AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END</pre>	<pre># кроме уже указанных # допускается использование # целочисленных переменных # m, k a = [] N = 2019 for i in range(0, N):     a.append(int(input())) ...</pre>

Алгоритмический язык	Паскаль
алг нач цел N=2019 целтаб a[1:N] цел i, m, k нц для i от 1 до N ввод a[i] кц ... Кон	const N=2019; var a: array [1..N] of integer; i, m, k: integer; begin for i:=1 to N do readln(a[i]); ... end.
<b>C++</b>	
<pre>#include &lt;iostream&gt; using namespace std; const int N=2019; int main(){     int a[N];     int i, m, k;     for (i=0; i&lt;N; ++i)         cin &gt;&gt; a[i];      ...     return 0; }</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 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Задача решается в два прохода: на первом проходе определяются минимум и максимум, после этого вычисляется их среднее, и на втором проходе производится корректировка и вывод элементов.</p> <p>Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод</p>
Пример правильной программы на языке Паскаль
<pre>m := 15001; k := 0; for i:=1 to N do begin     if (a[i] mod 2 = 0) and (a[i] &lt; m) then m := a[i];     if (a[i] mod 2 = 0) and (a[i] &gt; k) then k := a[i]; end; if m &gt; 15000 then m := 0; m := (m+k) div 2; for i:=1 to N do begin     if (a[i] mod 2 = 1) and (a[i] &gt; m) then         a[i] := a[i] - m;</pre>

```
writeln(a[i])
end;
```

При использовании языка Python для нахождения минимума и максимума можно применить функции `min` и `max`. При этом обязательно нужно использовать параметр `default` (доступен начиная с версии Python 3.4) или другим способом обеспечить обработку ситуации, когда в массиве нет чётных элементов

### Пример правильной программы на языке Python

```
m = min((i for i in a if i%2 == 0), default = 0)
k = max((i for i in a if i%2 == 0), default = 0)
m = (m+k) // 2
for i in range(0,N):
    if a[i]%2 == 1 and a[i] > m:
        a[i] -= m
print(a[i])
```

Использовать описанную возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на языке Паскаль

Указания по оцениванию	Баллы
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</p> <p>Не считается ошибкой и не приводит к снижению оценки использование для определения среднего значения обычной операции деления вместо деления нацело (в тех языках, где эти операции различаются).</p> <p>Эффективность не имеет значения и не оценивается.</p> <p>Допускается запись программы на языке, не входящем в список языков из условия. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если выбранный 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.</p> <p>Допускается изменение указанного в условии формата вывода, например, вывод всех элементов массива в одну строку</p>	
Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив	2
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку)</p>	1

1) Отсутствие инициализации или неверная инициализация минимума или максимума. 2) Неверное определение минимума или максимума. 3) Неверное определение среднего значения. 4) Неверное построение логических условий (поиск минимума или максимума на неверном множестве). 5) Выход за границы массива. 6) Исходный массив не изменяется. 7) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться. 8) Не учитывается или неверно обрабатывается ситуация, когда в исходном массиве нет чётных элементов. 9) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы). 10) Используется переменная, не объявленная в разделе описания переменных. 11) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно	
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
Максимальный балл	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один камень**, **увеличить** количество камней **в первой куче в два раза** или **увеличить** количество камней **во второй куче в три раза**. Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (12, 9), (6, 10), (6, 27). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 69. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 69 или больше камней.

В начальный момент в первой куче было 10 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 58$ .

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. В описание выигрышной стратегии не следует включать ходы играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

### Задание 1.

Назовите все значения  $S$ , при которых Петя может выиграть первым ходом.

### Задание 2.

Укажите такое значение  $S$ , при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанного значения  $S$  опишите выигрышную стратегию Пети.

### Задание 3.

Укажите такое значение  $S$ , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом 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). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

### Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

#### Задание 1.

Петя может выиграть первым ходом, если  $S = 20, \dots, 58$ . Для выигрыша достаточно утроить количество камней во второй куче. При меньших значениях  $S$  за один ход нельзя получить 69 или более камней в двух кучах.

#### Задание 2.

Возможные значения  $S$ : 16, 19. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при  $S = 16$  Петя может получить позицию (20, 16), а при  $S = 19$  – позицию (11, 19).

В первом случае после хода Вани возникнет одна из позиций (21, 16), (40, 16), (20, 17), (20, 48), во втором случае – одна из позиций (12, 19), (22, 19), (11, 20), (11, 57). В любой из перечисленных позиций Петя может выиграть, утроив количество камней во второй куче.

#### Задание 3.

Возможное значение  $S$ : 18. После первого хода Пети возможны позиции (11, 18), (20, 18), (10, 19), (10, 54). В позициях (20, 18) и (10, 64) Ваня может выиграть первым ходом, утроив количество камней во второй куче. Из позиций (11, 18) и (10, 19) Ваня может получить позицию (11, 19), разобранный в задании 2. Игрок, после хода которого возникла эта позиция (в данном случае – Ваня), выигрывает следующим ходом.

В таблице изображены возможные партии при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным

шрифтом. На рисунке эти же партии показаны в виде графа (оба способа изображения допустимы)

	Положения после очередных ходов			
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(10, 18) Всего 28	(20, 18) Всего 38	(20, 54) <b>Всего 74</b>		
	(10, 54) Всего 64	(10, 162) <b>Всего 172</b>		
	(11, 18) Всего 29	(11, 19) Всего 30	(12, 19) Всего 31	(12, 57) <b>Всего 69</b>
			(22, 19) Всего 41	(22, 57) <b>Всего 79</b>
	(10, 19) Всего 29		(11, 20) Всего 31	(11, 60) <b>Всего 71</b>
			(11, 57) Всего 68	(11, 171) <b>Всего 182</b>

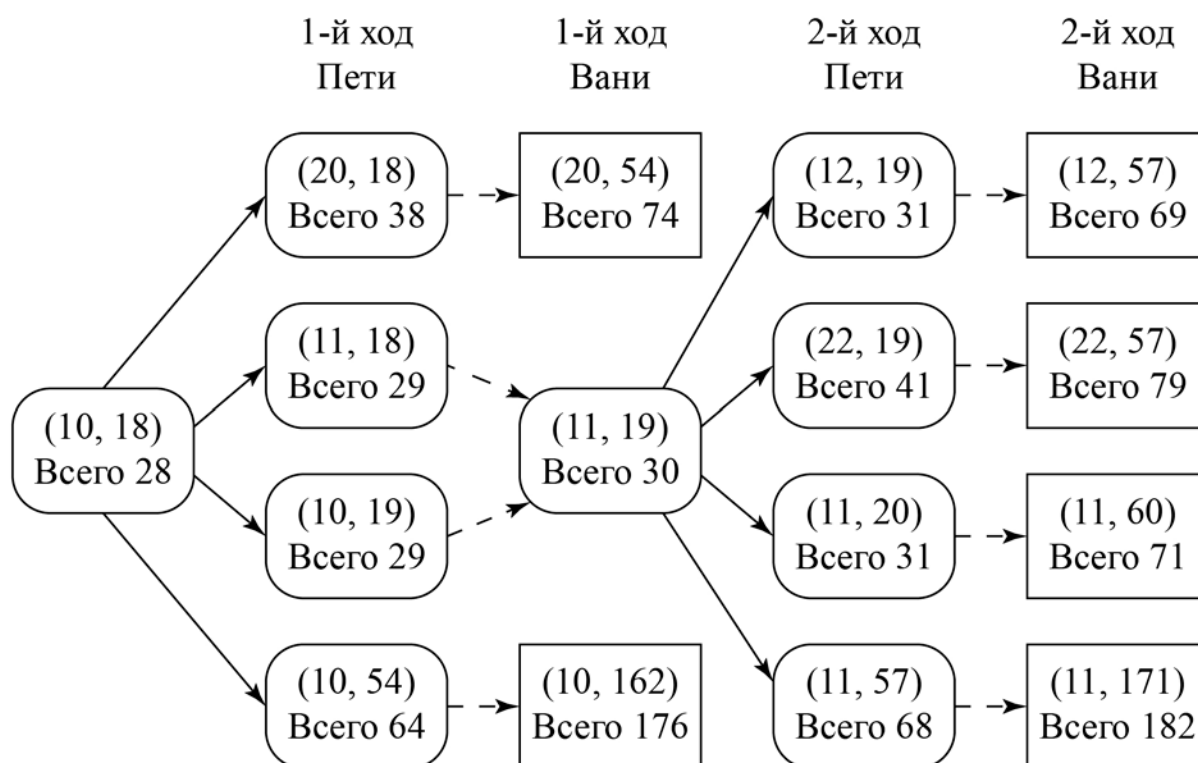




Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани показаны пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*Примечание для эксперта.* Дерево всех партий может быть изображено в виде таблицы или в виде ориентированного графа – так, как показано на рисунке, или другим способом. Например, вместо приведённого здесь «экономного» варианта, в котором позиции не дублируются, возможно построение полного дерева, в котором одинаковые позиции, возникающие при различном ходе игры, показаны отдельно. Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии со множеством партий, возможных при описанной в решении стратегии. В некоторых позициях заключительный выигрывающий ход можно сделать несколькими способами. В таблице и на рисунке указан один из них, в работе допускается выбор любого допустимого заключительного выигрывающего хода.

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить <b>три</b> задания. 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. Задание 1 выполнено, если перечислены все удовлетворяющие условию значения $S$ , и только они. Обоснование найденных значений не обязательно. Задание 2 выполнено, если верно указана выигрышная для Пети позиция (любая из двух возможных) и описана соответствующая стратегия. Задание 3 выполнено, если правильно указана выигрышная для Вани позиция и построено дерево всех возможных при выигрышной стратегии партий (и только их). Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены все три задания	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. – Выполнено задание 3. – Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из заданий 1 и 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

**27**

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, находящихся на расстоянии не меньше 6 (разница в индексах элементов должна быть 6 или более). Необходимо определить количество пар, разность чисел в которых кратна 3.

**Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $6 \leq N \leq 1000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

8  
2  
6  
5  
4  
6  
7  
9  
8

*Пример выходных данных для приведённого выше примера входных данных:*

1

*Пояснение.* Из восьми чисел можно составить три пары, удовлетворяющие условию. Это будут элементы с индексами 1 и 7, 1 и 8, 2 и 8. Для заданного набора чисел получаем пары (2, 9), (2, 8), (6, 8). Разности чисел в этих парах равны 7, 6, 2. Одна из этих разностей кратна 3.

Напишите эффективную по времени и по памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает одного килобайта и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **бóльшая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

<p align="center"><b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Разность двух чисел кратна 3, если остатки от деления этих чисел на 3 равны. Будем хранить количество чисел, дающих при делении на 3 остатки 0, 1 и 2 без учёта шести последних элементов последовательности. Для этого понадобится хранить последние шесть элементов. Остальные элементы последовательности можно не хранить, это обеспечивает эффективность по памяти. Для хранения шести элементов можно использовать циклический массив, как показано в решении 1.</p> <p>Будем рассматривать каждое введённое число как правый элемент возможной пары (первые шесть чисел не могут быть таким элементом) и в зависимости от остатка при его делении на 3 определять количество подходящих пар и прибавлять его к общей сумме</p>
<p><b>Решение 1. Правильная и эффективная программа на языке Паскаль (циклический массив, счётчики в отдельном массиве)</b></p>
<pre>const d=6; {требуемое расстояние между элементами} var   N: integer;      {количество чисел}   x: integer;      {очередное число}   a: array[0..d-1] of integer;   k: array[0..2] of integer; {количество по остаткам}   s: integer;      {количество подходящих пар}   i: integer;      {счётчик для ввода}   ia: integer;     {текущий индекс в массиве a} begin   readln(N);   {ввод первых d чисел}   for i:=0 to d-1 do readln(a[i]);   {ввод и обработка остальных значений}   for i:=0 to 2 do k[i]:=0;   ia:=0; s:=0;   for i:=d to N-1 do begin     readln(x);     k[a[ia] mod 3] := k[a[ia] mod 3] + 1;     s := s + k[x mod 3];     a[ia] := x;     ia := (ia+1) mod d   end;   writeln(s) end.</pre>
<p>Вместо циклического массива можно использовать сдвиги. В этом случае для вычисления всегда используется первый элемент массива, а новое число</p>

записывается в последний. Хотя этот алгоритм работает медленнее, чем алгоритм с циклическим массивом (для каждого элемента требуется пять дополнительных присваиваний при сдвигах), основное требование эффективности здесь выполнено: при увеличении размера массива в  $k$  раз количество действий растёт не более чем в  $k$  раз.

Для хранения счётчиков по остаткам можно вместо массива из трёх элементов использовать три отдельные переменные. Это делает программу чуть более громоздкой, но не влияет на эффективность.

Ниже приводится пример программы, в которой использованы эти методы

## **Решение 2. Правильная и эффективная программа на языке Паскаль (сдвиг массива, счётчики в отдельных переменных)**

```
const d=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[1..d] of integer;
  k0, k1, k2: integer; {количество по остаткам}
  s: integer;      {количество подходящих пар}
  i: integer;      {счётчик для ввода}
  ia: integer;     {счётчик для сдвига}
begin
  readln(N);
  {ввод первых d чисел}
  for i:=1 to d do readln(a[i]);
  {ввод и обработка остальных значений}
  k0 := 0; k1 := 0; k2:=0;
  s:=0;
  for i:=d+1 to N do begin
    readln(x);
    if a[1] mod 3 = 0 then k0 := k0 + 1;
    if a[1] mod 3 = 1 then k1 := k1 + 1;
    if a[1] mod 3 = 2 then k2 := k2 + 1;
    if x mod 3 = 0 then s := s + k0;
    if x mod 3 = 1 then s := s + k1;
    if x mod 3 = 2 then s := s + k2;
    for ia:=1 to s-1 do a[ia]:=a[ia+1];
    a[s] := x;
  end;
  writeln(s)
end.
```

Возможно также «любовое» решение: запишем все исходные числа в массив, переберём все возможные пары и подсчитаем количество подходящих. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растёт квадратично). Такая программа оценивается не выше 2 баллов.

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)

**Решение 3. Правильная, но неэффективная программа на языке Паскаль**

```

const d=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  a: array [1..1000] of integer; {исходные данные}
  s: integer;      {количество подходящих пар}
  i,j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  s :=0;
  for i := 1 to N-d do begin
    for j := i+d to N do begin
      if a[i] mod 3 = a[j] mod 3
        then s := s +1
    end
  end;
  writeln(s)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается в 0 баллов	
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> <li>1) пропущен или неверно указан знак пунктуации;</li> <li>2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</p>	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</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются</p>	3

<p>в массиве, контейнере STL в C++ или другой аналогичной структуре данных).</p> <p>Количество синтаксических ошибок (описок), указанных в критериях на 4 балла, – не более пяти.</p> <p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (не считается значение <math>N</math> или неверно организован ввод последовательности);</li> <li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li> <li>3) используется неверный тип данных;</li> <li>4) использована одна переменная (константа) вместо другой;</li> <li>5) используется один знак операции вместо другого;</li> <li>6) отсутствует вывод ответа или выводится не то значение (хотя правильный ответ в программе найден);</li> <li>7) неверная работа с массивом, в том числе выход за границы массива;</li> <li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками)</li> </ol>	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.</p> <p>ИЛИ</p> <p>Программа работает в целом верно и эффективно по времени, но в ней есть не подходящие под перечень из критериев на 3 балла ошибки, которые в некоторых особых случаях приводят к неверным результатам.</p> <p>ИЛИ</p> <p>Представлено корректное переборное решение, в котором все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например, выход индексов за границы массива, неверный учёт расстояния между элементами и т. д.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа представлена и содержит как минимум два обязательных элемента, возможно, реализованных с ошибками:</p> <ol style="list-style-type: none"> <li>1) рассматриваются только пары, находящиеся на расстоянии не меньше заданного в условии;</li> <li>2) ведётся подсчёт пар с подходящей разностью</li> </ol>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p style="text-align: right;"><i>Максимальный балл</i></p>	4