

### Критерии оценивания заданий с развёрнутым ответом

24

Дано целое положительное число  $N$ . Необходимо найти и вывести количество и сумму чётных цифр в десятичной записи  $N$ . Если чётных цифр в числе нет, нужно вывести слово "NO".

Например, для  $N = 2018$  нужно вывести числа 3 (в числе три чётные цифры) и 10 ( $2 + 0 + 8 = 10$ ), а для  $N = 1993$  – слово "NO".

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, D, K, S AS INTEGER INPUT N K = 0 S = 0 WHILE N &gt; 1     D = N MOD 10     IF D MOD 2 = 0 THEN         K = K + 1         S = D     END IF     N = N \ 10 WEND IF S &gt; 0 THEN     PRINT K, S ELSE     PRINT "NO" END IF </pre>	<pre> n = int(input()) k = 0 s = 0 while n &gt; 1:     d = n % 10     if d % 2 == 0:         k += 1         s = d     n = n // 10 if s &gt; 0:     print(k, s) else:     print("NO") </pre>

Алгоритмический язык	Паскаль
<pre> алг нач     цел n, d, k, s     ввод n     k := 0     s := 0     нц пока n &gt; 1         d := mod(n,10)         если mod(d,2)=0 то             k := k + 1             s := d         все         n := div(n,10)     кц     если s &gt; 0 то         вывод k, ' ', s     иначе         вывод "NO"     все кон </pre>	<pre> var n, d, k, s: integer; begin     readln(n);     k := 0;     s := 0;     while n &gt; 1 do begin         d := n mod 10;         if d mod 2 = 0 then begin             k := k + 1;             s := d         end;         n := n div 10     end;     if s &gt; 0 then         write(k, ' ', s)     else         write('NO')     end. </pre>
<b>C++</b>	
<pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, d, k, s;     cin &gt;&gt; n;     k = 0;     s = 0;     while (n &gt; 1) {         d = n % 10;         if (d % 2 == 0) {             ++k;             s = d;         }         n /= 10;     }     if (s &gt; 0)         cout &lt;&lt; k &lt;&lt; ' ' &lt;&lt; s;     else         cout &lt;&lt; "NO";     return 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе  $N = 2018$ .
2. Приведите пример трёхзначного числа  $N$ , содержащего хотя бы одну чётную цифру, при вводе которого программа выведет верный ответ.
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)
<p>1. При вводе <math>N = 2018</math> программа выведет числа 3 и 2.</p> <p>2. Программа выводит верный ответ, если в числе ровно одна ненулевая чётная цифра и эта цифра расположена левее всех нулей. Примеры таких трёхзначных чисел: 114, 360, 817, 200.</p> <p>3. Программа содержит две ошибки.</p> <p><b>Первая ошибка.</b> Неверный подсчёт суммы цифр. Очередная найденная цифра не прибавляется к сумме, а записывается в неё. В результате вместо суммы чётных цифр определяется первая слева чётная цифра.</p> <p><b>Вторая ошибка.</b> Неверное условие наличия чётных цифр. Вместо количества цифр проверяется их сумма. В результате в ситуации, когда в числе есть нули, но нет других чётных цифр, будет выведен неверный ответ.</p> <p><b>Не является ошибкой</b> условие цикла. Проверка <math>n &gt; 1</math> вместо <math>n &gt; 0</math> приводит к тому, что если число начинается с единицы, то эта единица не проверяется, но поскольку единица не является чётной цифрой, это не влияет на правильность ответа.</p> <p><b>Пример исправления для алгоритмического языка</b></p> <p><b>Первая ошибка:</b></p> <pre>s := d</pre> <p>Исправленная строка:</p> <pre>s := s + d</pre> <p><b>Вторая ошибка:</b></p> <pre>если s &gt; 0 то</pre> <p>Исправленная строка:</p> <pre>если k &gt; 0 то</pre> <p>В программах на других языках ошибочные строки и их исправления аналогичны</p>

Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков, не влияющих на содержательную часть исправления (например, слова «то» 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, что выведет программа при конкретном вводе.</li> <li>2. Указать пример входного числа, при котором программа выдаёт 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 указан верный результат работы программы при заданном входном значении.</p> <p>Действие 2 считается выполненным, если указано входное значение и при выполнении программы с этим входным значением получается ответ, совпадающий с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Для действий 1 и 2 учащийся 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> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</li> </ol> <p>В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка)</p>	
Выполнены все четыре необходимых действия, 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> <ol style="list-style-type: none"> <li>1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной.</li> <li>2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной</li> </ol>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла.</p> <p>Выполнены два из четырёх необходимых действий</p>	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 количество нечётных элементов массива, кратных 3, заменить все чётные элементы, не кратные 3, на это количество и вывести изменённый массив. Например, для исходного массива из пяти элементов 33, 89, 27, 92, 48 программа должна вывести числа 33, 89, 27, 2, 48.

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
<pre>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</pre>	<pre>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.</pre>
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.

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Задача решается в два прохода: на первом проходе определяется количество требуемых элементов, 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>	
<b>Пример правильной программы на алгоритмическом языке</b>	
<pre> к := 0 нц для i от 1 до N   если mod(a[i],2)=1 и mod(a[i],3)=0     то к :=к + 1   все кц нц для i от 1 до N   если mod(a[i],2)=0 и mod(a[i],3)&gt;0     то a[i] := к   все   вывод a[i], ' ' кц </pre>	
При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка	
<b>Пример правильной программы на языке Python</b>	
<pre> k = sum(1 for i in a if i%2 == 1 and i%3 == 0) for i in range(0,N):     if a[i]%2 == 0 and a[i]%3 &gt; 0:         a[i] = k     print(a[i]) </pre>	
Использовать описанную возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на алгоритмическом языке	

<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</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,</p>	

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. Допускается произвольный формат вывода полученного массива, например вывод всех элементов массива в одну строку или вывод каждого элемента в отдельной строке. Отмечается как ошибка, но не учитывается при выставлении оценки вывод элементов в одну строку без пробелов между ними	
Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив	2
Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку). 1) Отсутствие инициализации или неверная инициализация счётчика. 2) Неверное определение чётности и кратности 3. 3) Неверное построение логических условий (неверные логические операции, проверка не всех условий). 4) Выход за границы массива. 5) Исходный массив не изменяется. 6) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться. 7) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы). 8) Используется переменная, не объявленная в разделе описания переменных. 9) Индексная переменная в цикле не меняется (например, в цикле <code>while</code> ) или меняется неверно	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	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** или **увеличить** количество камней **в 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), (18, 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 75. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 75 или больше камней.

В начальный момент в первой куче было 7 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.
- Петя сделал неудачный первый ход, после которого Ваня выиграл своим первым ходом. Назовите минимальное значение  $S$ , при котором это возможно.

### 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)	
<b>Задание 1.</b>	
а) 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 75 или более камней в двух кучах.	
б) Такая ситуация возможна при $S = 8$ . Если Петя утроит вторую кучу,	



получится позиция (7, 24), из которой Ваня может получить позицию (7, 72) и выиграть. При  $S < 8$  никакой первый ход Пети не создаст ситуацию, в которой Ваня может сразу выиграть.

### Задание 2.

Возможные значения  $S$ : 11, 22. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при  $S = 11$  Петя может получить позицию (21, 11), а при  $S = 22$  – позицию (8, 22).

В первом случае после хода Вани возникнет одна из позиций (22, 11), (63, 11), (21, 12), (21, 33), во втором случае – одна из позиций (9, 22), (24, 22), (8, 23), (8, 66). В любой из перечисленных позиций Петя может выиграть, утроив количество камней в большей куче.

### Задание 3.

Возможное значение  $S$ : 21. После первого хода Пети возможны позиции (8, 21), (21, 21), (7, 22), (7, 63). В позициях (21, 21) и (7, 63) Ваня может выиграть первым ходом, утроив количество камней во второй куче. Из позиций (8, 21) и (7, 22) Ваня может получить позицию (8, 22), разобрannую 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)
(7, 21) Всего 28	(21, 21) Всего 42	(21, 63) <b>Всего 84</b>		
	(7, 63) Всего 70	(7, 189) <b>Всего 196</b>		
	(8, 21) Всего 29	(8, 22) Всего 30	(9, 22) Всего 31	(9, 66) <b>Всего 75</b>
			(24, 22) Всего 46	(24, 66) <b>Всего 90</b>
	(7, 22) Всего 29		(8, 23) Всего 31	(8, 69) <b>Всего 77</b>
			(8, 66) Всего 74	(8, 192) <b>Всего 200</b>

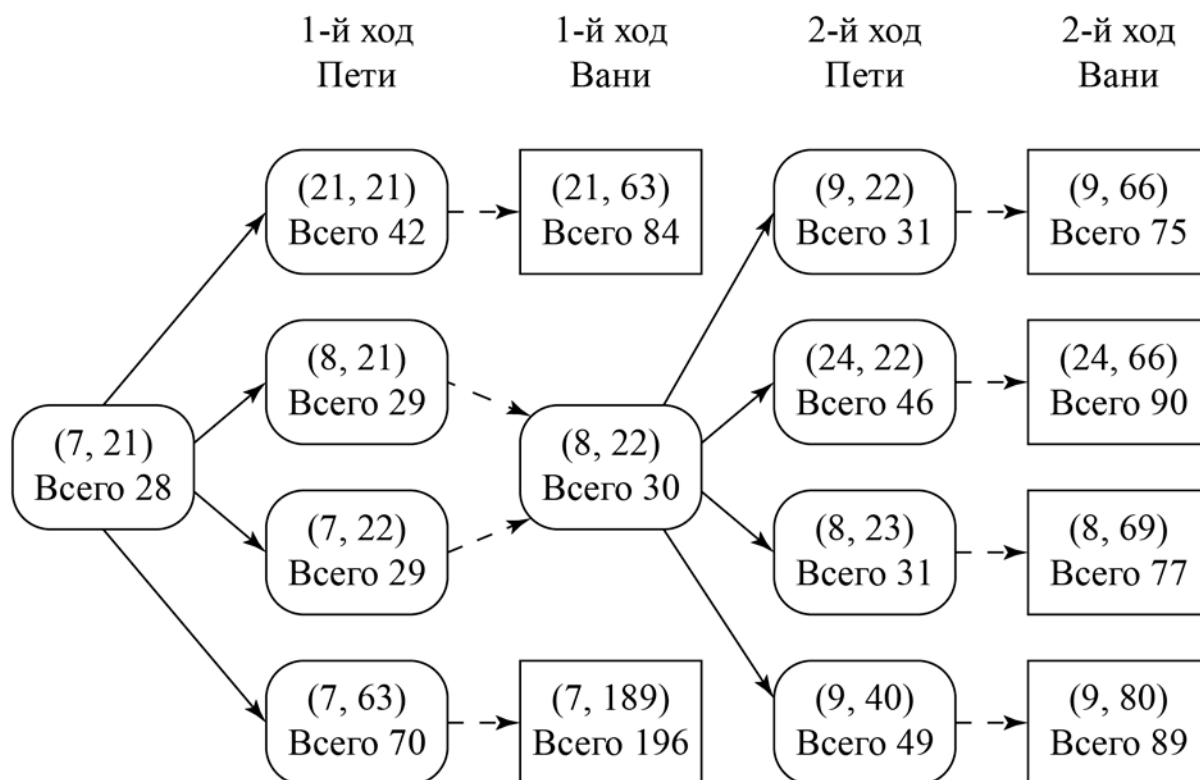


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*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 с 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>	

Задание 1 выполнено, если выполнены оба пункта: для пункта (а) перечислены все удовлетворяющие условию значения $S$ , и только они; для пункта (б) указано верное значение $S$ , и только оно. 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, сумма которых делится на  $m = 80$ . Среди всех таких пар нужно найти и вывести пару с максимальным произведением элементов. Если одинаковое максимальное произведение имеют несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

8  
10  
30  
50  
40  
60  
70  
90  
80

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

70 90

*Пояснение.* Из данных восьми чисел можно составить три пары, удовлетворяющие условию: (10, 70), (30, 50), (70, 90). Наибольшее произведение получается в паре (70, 90).

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

Программа считается эффективной по времени, если при одновременном увеличении количества исходных чисел  $N$  и параметра  $m$  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 4 Кбайт 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  $m$ , если сумма остатков от деления этих чисел на  $m$  кратна  $m$ . При этом для получения максимального произведения из чисел с одинаковыми остатками нужно выбирать наибольшее.

Будем хранить в массиве из  $m$  элементов максимальное число, имеющее соответствующий остаток от деления на  $m$ .

Каждое введённое число будем рассматривать как правый элемент возможной пары, находить соответствующий ему парный остаток, вычислять произведение пары и при необходимости обновлять максимум для произведения и для чисел с данным остатком.

Ниже приведена программа на алгоритмическом языке, реализующая этот алгоритм

**Решение 1. Правильная и эффективная программа на алгоритмическом языке**

```

алг задача27
нач
    цел m = 80
    цел таб a[0:m-1]
    цел N
    цел x | очередное число из последовательности
    цел x1, x2 | ответ – пара чисел
    цел p, pp | остаток и парный ему остаток
    цел i

    нц для i от 0 до m-1
        a[i] := 0
    кц
    x1 := 0; x2 := 0
    ввод N
    нц N раз
        ввод x
        p := mod(x, m)
        pp := mod(m-p, m)
        если x*a[pp] > x1*x2
            то x1 := a[pp]; x2 := x
        все
        если x > a[p]
            то a[p] := x
        все
    кц
    вывод x1, ' ', x2
кон

```

В приведённом решении в переменную  $p$  записывается остаток от деления на  $m$  очередного числа из последовательности, а в переменную  $pp$  – парный к нему остаток, то есть тот остаток, который должен быть у второго числа пары, чтобы сумма чисел делилась на  $m$ . При вычислении учитывается особый случай, когда заданное число делится на  $m$ , то есть  $p = 0$ . В этом случае парный остаток тоже должен быть равен 0. Возможно вычисление  $pp$  с использованием конструкции «если» или обработка этой ситуации в отдельной ветке программы. Во всех этих случаях (если они реализованы без ошибок) программа остаётся правильной и эффективной и может быть оценена высшим баллом.

Порядок проверки условий (сначала обновление произведения, затем – элемента массива) важен. Если сначала обновить массив, то при  $p = 0$  и  $p = m/2$  вместо произведения двух разных элементов может быть получен квадрат одного элемента последовательности.

В программе нет специальной проверки наличия в последовательности подходящих пар, но если их нет, переменные  $x1$  и  $x2$  сохранят заданные в начале программы нулевые значения и будет выведен верный ответ.

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и выберем подходящую. 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). Подобная программа оценивается не выше 2 баллов.

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

## Решение 2. Правильная, но неэффективная программа на языке Паскаль

```
const m=80;
var
  N: integer;      {количество чисел}
  a: array [1..10000] of integer; {исходные данные}
  x1, x2: integer; {ответ - пара чисел}
  i, j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  x1:=0; x2:=0;
  for i := 1 to N do begin
    for j := i+1 to N do begin
      if ((a[i] + a[j]) mod m = 0) and
        (a[i]*a[j] > x1*x2)
      then begin
        x1:=a[i]; x2:=a[j]
      end
    end
  end;
  writeln(x1, ' ', x2)
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	
Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел $N$ и параметра $m$ , время работы пропорционально $N$ . Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов: 1) пропущен или неверно указан знак пунктуации; 2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования;	4

<p>3) не описана или неверно описана переменная;</p> <p>4) применяется операция, недопустимая для соответствующего типа данных.</p> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку</p>	
<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 <math>N</math>, но может зависеть от <math>m</math>.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве 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 (например, выводится произведение вместо пары чисел);</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> <li>9) неверно обрабатываются элементы, кратные <math>m</math></li> </ol>	3
<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 style="text-align: center;">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 style="text-align: center;">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</p>	2

индексов за границы массива, рассмотрение произведений вида $a[i]*a[i]$ и т. д.	
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: 1) рассматриваются пары с подходящей суммой; 2) обнаруживается и выводится пара с максимальным произведением	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 и вывести количество и сумму цифр, кратных 3, в десятичной записи  $N$ . Если кратных 3 цифр в числе нет, нужно вывести слово "NO".

Например, для  $N = 1960$  нужно вывести числа 3 (в числе три кратные 3 цифры) и 15 ( $9 + 6 + 0 = 15$ ), а для  $N = 1812$  – слово "NO".

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

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

<b>Бейсик</b>	<b>Python</b>
<pre> DIM N, D, K, S AS INTEGER INPUT N K = 0 S = 0 WHILE N &gt; 10     D = N MOD 10     IF D MOD 3 = 0 THEN         K = K + 1         S = S + D     END IF     N = N \ 10; WEND IF S &gt; 0 THEN     PRINT K, S ELSE     PRINT "NO" END IF </pre>	<pre> n = int(input()) k = 0 s = 0 while n &gt; 10:     d = n % 10     if d % 3 == 0:         k += 1         s += d     n = n // 10 if s &gt; 0:     print(k, s) else:     print("NO") </pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач     цел n, d, k, s     ввод n     k := 0     s := 0     нц пока n &gt; 10         d := mod(n, 10)         если mod(d, 3) = 0 то             k := k + 1             s := s + d         все         n := div(n, 10)     кц     если s &gt; 0 то         вывод k, ' ', s     иначе         вывод "NO" </pre>	<pre> var n, d, k, s: integer; begin     readln(n);     k := 0;     s := 0;     while n &gt; 10 do begin         d := n mod 10;         if d mod 3 = 0 then begin             k := k + 1;             s := s + d;         end;         n := n div 10;     end;     if s &gt; 0 then         write(k, ' ', s)     else         write('NO')     end. </pre>

Всё КОН	
<b>C++</b>	
<pre> #include &lt;iostream&gt; using namespace std; int main() {     int n, d, k, s;     cin &gt;&gt; n;     k = 0;     s = 0;     while (n &gt; 10) {         d = n % 10;         if (d % 3 == 0) {             ++k;             s += d;         }         n /= 10;     }     if (s &gt; 0)         cout &lt;&lt; k &lt;&lt; ' ' &lt;&lt; s;     else         cout &lt;&lt; "NO";     return 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе  $N = 3961$ .
2. Приведите пример трёхзначного числа  $N$ , содержащего хотя бы одну кратную 3 цифру, при вводе которого программа выведет верный ответ.
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. При вводе <math>N = 3961</math> программа выведет числа 2 и 15.</li> <li>2. Программа выводит верный ответ, если первая цифра исходного числа не кратна 3 и в числе есть ненулевые кратные 3 цифры. Примеры таких трёхзначных чисел: 116, 260, 739, 803</li> <li>3. Программа содержит две ошибки.</li> </ol>

**Первая ошибка.** Неверное условие цикла. Первая цифра числа не проверяется, и если она кратна 3, получается неверный результат.

**Вторая ошибка.** Неверное условие наличия кратных 3 цифр. Вместо количества цифр проверяется их сумма. В результате в ситуации, когда в числе есть нули, но нет других кратных 3 цифр, будет выведен неверный ответ.

### Пример исправления для алгоритмического языка

#### Первая ошибка:

нц пока  $n > 10$

Исправленная строка:

нц пока  $n > 0$

Допустимы также следующие сравнения, при которых первая цифра числа проверяется не всегда, но цифры, кратные 3, будут обязательно учтены:

нц пока  $n > 1$

нц пока  $n > 2$

нц пока  $n \geq 1$

нц пока  $n \geq 2$

нц пока  $n \geq 3$

#### Вторая ошибка:

если  $s > 0$  то

Исправленная строка:

если  $k > 0$  то

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, слов «нц» или «то»)

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать, что выведет программа при конкретном вводе.</li> <li>2. Указать пример входного числа, при котором программа выдаёт 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 указан верный результат работы программы при заданном входном значении.</p> <p>Действие 2 считается выполненным, если указано входное значение и при выполнении программы с этим входным значением получается ответ, совпадающий с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Для действий 1 и 2 учащийся 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>	<i>3</i>

25

Дан массив, содержащий 2019 положительных целых чисел, не превышающих 15 000. Необходимо найти количество чётных элементов массива, не кратных 3, заменить все нечётные элементы, кратные 3, на это количество и вывести изменённый массив. Например, для исходного массива из пяти элементов 20, 89, 27, 92, 48 программа должна вывести числа 20, 89, 2, 92, 48.

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.

<b>Бейсик</b>	<b>Python</b>
<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
<pre> 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 </pre>	<pre> 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. </pre>
<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 определяется количество требуемых элементов, 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 алгоритмическом языке
<pre> к := 0 нц для i от 1 до N     если mod(a[i],2)=0 и mod(a[i],3)&gt;0         то к :=к + 1     все кц нц для i от 1 до N     если mod(a[i],2)=1 и mod(a[i],3)=0         то a[i] := к </pre>

<pre> все вывод a[i], ' ' кц </pre>
<p>При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка</p>
<p><b>Пример правильной программы на языке Python</b></p> <pre> k = sum(1 for i in a if i%2 == 0 and i%3 &gt; 0) for i in range(0,N):     if a[i]%2 == 1 and a[i]%3 == 0:         a[i] = k print(a[i]) </pre>
<p>Использовать описанную возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на алгоритмическом языке</p>

Указания по оцениванию	Баллы
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</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 произвольный формат вывода полученного массива, например вывод всех элементов массива в одну строку или вывод каждого элемента в отдельной строке. Отмечается как ошибка, но не учитывается при выставлении оценки вывод элементов в одну строку без пробелов между ними</p>	
<p>Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив</p>	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> <ol style="list-style-type: none"> <li>1) Отсутствие инициализации или неверная инициализация счётчика.</li> <li>2) Неверное определение чётности и кратности 3.</li> <li>3) Неверное построение логических условий (неверные логические операции, проверка не всех условий).</li> <li>4) Выход за границы массива.</li> </ol>	1

5) Исходный массив не изменяется. 6) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться. 7) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы). 8) Используется переменная, не объявленная в разделе описания переменных. 9) Индексная переменная в цикле не меняется (например, в цикле <code>while</code> ) 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** или **увеличить** количество камней **в 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), (18, 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 88. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 88 или больше камней.

В начальный момент в первой куче было 8 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 79$ .

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.
- Петя сделал неудачный первый ход, после которого Ваня выиграл своим первым ходом. Назовите минимальное значение  $S$ , при котором это возможно.

**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.**

а) Петя может выиграть первым ходом, если  $S = 27, \dots, 79$ . Для выигрыша достаточно утроить количество камней во второй куче. При меньших значениях  $S$  за один ход нельзя получить 88 или более камней в двух кучах.

б) Такая ситуация возможна при  $S = 9$ . Если Петя утроит вторую кучу, получится позиция  $(8, 27)$ , из которой Ваня может получить позицию  $(8, 81)$  и выиграть. При  $S < 9$  никакой первый ход Пети не создаст ситуацию, в которой Ваня может сразу выиграть.

**Задание 2.**

Возможные значения  $S$ : 15, 26. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при  $S = 15$  Петя может получить позицию  $(24, 15)$ , а при  $S = 26$  – позицию  $(9, 26)$ .

В первом случае после хода Вани возникнет одна из позиций  $(25, 15)$ ,  $(72, 15)$ ,  $(24, 16)$ ,  $(24, 45)$ , во втором случае – одна из позиций  $(10, 26)$ ,  $(27, 26)$ ,  $(9, 27)$ ,  $(9, 78)$ . В любой из перечисленных позиций Петя может выиграть, утроив количество камней в большей куче.

**Задание 3.**

Возможное значение  $S$ : 25. После первого хода Пети возможны позиции  $(9, 25)$ ,  $(24, 25)$ ,  $(8, 26)$ ,  $(8, 75)$ . В позициях  $(24, 25)$  и  $(8, 75)$  Ваня может выиграть первым ходом, утроив количество камней во второй куче. Из позиций  $(9, 25)$  и  $(8, 26)$  Ваня может получить позицию  $(9, 26)$ , 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)
(8, 25) Всего 33	(24, 25) Всего 49	(24, 75) <b>Всего 99</b>		
	(8, 75) Всего 83	(8, 225) <b>Всего 233</b>		
	(9, 25) Всего 34	(9, 26) Всего 35	(10, 26) Всего 36	(10, 78) <b>Всего 88</b>
			(27, 26) Всего 53	(81, 26) <b>Всего 107</b>
	(8, 26) Всего 34		(9, 27) Всего 36	(9, 81) <b>Всего 90</b>
			(9, 78) Всего 87	(9, 224) <b>Всего 233</b>
	1-й ход Пети	1-й ход Вани	2-й ход Пети	2-й ход Вани

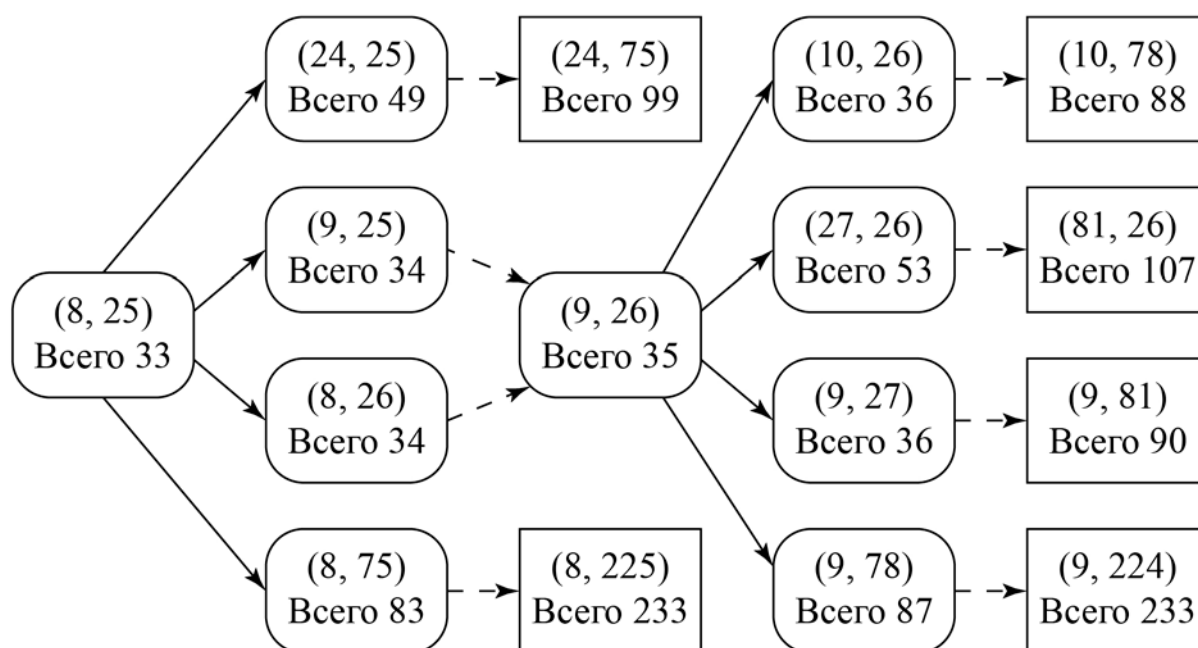


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*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 с 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 выполнены оба пункта: для пункта (а) перечислены все удовлетворяющие условию значения <math>S</math>, и только они; для пункта (б) указано верное значение <math>S</math>, и только оно. 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, сумма которых делится на  $m = 60$ . Среди всех таких пар нужно найти и вывести пару с максимальным произведением элементов. Если одинаковое максимальное произведение имеют несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

**Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

8  
10  
30  
50  
40  
60  
70  
90  
80

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

50 70

*Пояснение.* Из данных восьми чисел можно составить четыре пары, удовлетворяющие условию: (10, 50), (30, 90), (50, 70), (40, 80). Наибольшее произведение получается в паре (50, 70).

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

Программа считается эффективной по времени, если при одновременном увеличении количества исходных чисел  $N$  и параметра  $m$  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 4 Кбайт 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>Сумма двух чисел кратна <math>m</math>, если сумма остатков от деления этих чисел на <math>m</math> кратна <math>m</math>. При этом для получения максимального произведения из чисел с одинаковыми остатками нужно выбирать наибольшее.</p> <p>Будем хранить в массиве из <math>m</math> элементов максимальное число, имеющее соответствующий остаток от деления на <math>m</math>.</p> <p>Каждое введённое число будем рассматривать как правый элемент возможной пары, находить соответствующий ему парный остаток, вычислять произведение пары и при необходимости обновлять максимум для произведения и для чисел с данным остатком.</p> <p>Ниже приведена программа на алгоритмическом языке, реализующая этот алгоритм</p>
<p><b>Решение 1. Правильная и эффективная программа на алгоритмическом языке</b></p>
<pre> алг задача27 нач   цел m = 60   цел таб a[0:m-1]   цел N   цел x   очередное число из последовательности   цел x1,x2   ответ - пара чисел   цел p,pp   остаток и парный ему остаток   цел i    нц для i от 0 до m-1     a[i] := 0   кц   x1 := 0; x2 := 0   ввод N   нц N раз     ввод x     p := mod(x,m)     pp := mod(m-p,m)     если x*a[pp] &gt; x1*x2       то x1 := a[pp]; x2 := x     все     если x &gt; a[p]       то a[p] := x     все   кц   вывод x1, ' ', x2 конец </pre>

В приведённом решении в переменную  $p$  записывается остаток от деления на  $m$  очередного числа из последовательности, а в переменную  $pp$  – парный к нему остаток, то есть тот остаток, который должен быть у второго числа пары, чтобы сумма чисел делилась на  $m$ . При вычислении учитывается особый случай, когда заданное число делится на  $m$ , то есть  $p = 0$ . В этом случае парный остаток тоже должен быть равен 0. Возможно вычисление  $pp$  с использованием конструкции «если» или обработка этой ситуации в отдельной ветке программы. Во всех этих случаях (если они реализованы без ошибок) программа остаётся правильной и эффективной и может быть оценена высшим баллом.

Порядок проверки условий (сначала обновление произведения, затем – элемента массива) важен. Если сначала обновить массив, то при  $p = 0$  и  $p = m/2$  вместо произведения двух разных элементов может быть получен квадрат одного элемента последовательности.

В программе нет специальной проверки наличия в последовательности подходящих пар, но если их нет, переменные  $x1$  и  $x2$  сохраняют заданные в начале программы нулевые значения и будет выведен верный ответ.

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и выберем подходящую. 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). Подобная программа оценивается не выше 2 баллов.

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

### Решение 2. Правильная, но неэффективная программа на языке Паскаль

```
const m=60;
var
  N: integer;      {количество чисел}
  a: array [1..10000] of integer; {исходные данные}
  x1, x2: integer; {ответ – пара чисел}
  i, j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  x1:=0; x2:=0;
  for i := 1 to N do begin
    for j := i+1 to N do begin
      if ((a[i] + a[j]) mod m = 0) and
        (a[i]*a[j] > x1*x2)
      then begin
        x1:=a[i]; x2:=a[j]
      end
    end
  end;
  writeln(x1, ' ', x2)
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 <math>N</math> и параметра <math>m</math>, время работы пропорционально <math>N</math>.</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 <math>N</math>, но может зависеть от <math>m</math>.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве 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 (например, выводится произведение вместо пары чисел);</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> <li>9) неверно обрабатываются элементы, кратные <math>m</math></li> </ol>	3

<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 style="text-align: center;">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 style="text-align: center;">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, рассмотрение произведений вида <math>a[i]*a[i]</math> 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 пары с подходящей суммой;</li> <li>2) обнаруживается и выводится пара с максимальным произведением</li> </ol>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p><i>Максимальный балл</i></p>	4