

### Критерии оценивания заданий с развёрнутым ответом

24

Даны 4 целых положительных числа. Необходимо выбрать из них и вывести на экран число с наибольшей последней цифрой. Если в наборе несколько чисел с одинаковой наибольшей последней цифрой, нужно вывести наибольшее из этих чисел.

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 M, X, DX AS INTEGER M = 0 FOR I = 1 to 4     INPUT X     DX = X MOD 10     IF DX &gt;= M MOD 10 THEN         M = X     ELSE         IF DX=M MOD 10 AND X&gt;M THEN             M = DX         END IF     END IF NEXT I PRINT M </pre>	<pre> m = 0 for i in range (4):     x = int(input())     dx = x % 10     if dx &gt;= m%10:         m = x     else:         if dx == m%10 and x&gt;m:             m = dx print(m) </pre>
C++	Паскаль
<pre> #include &lt;iostream&gt; using namespace std; int main() {     int m, x, dx, i;     m = 0;     for (i=1; i&lt;=4; ++i) {         cin &gt;&gt; x;         dx = x%10;         if (dx &gt;= m%10)             m = x;         else {             if (dx==m%10 &amp;&amp; x&gt;m)                 m = dx;         }     }     cout &lt;&lt; m;     return 0; } </pre>	<pre> var m, x, dx, i: integer; begin     m := 0;     for i:=1 to 4 do begin         readln(x);         dx := x mod 10;         if dx &gt;= m mod 10             then m :=x         else begin             if (dx=m mod 10) and (x&gt;m)                 then m :=dx             end         end;         write(m)     end. end. </pre>

**Алгоритмический язык**

```

алг
нач
  цел m, x, dx, i
  m := 0
  для i от 1 до 4
    ввод x
    dx := mod(x,10)
    если dx >= mod(m,10)
      то m := x
    иначе
      если dx = mod(m,10) и x>m
        то m := dx
      все
    все
  кц
  вывод m
кон

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 12, 13, 104, 24.
2. Приведите пример исходных данных, при вводе которых программа, несмотря на ошибки, 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)

1. При вводе указанных чисел программа выведет число 24.
2. Программа выводит верный ответ А, если после ввода А не вводится меньшее число с той же последней цифрой. Пример такого набора: 15, 20, 37, 32.
3. Программа содержит две ошибки.

**Первая ошибка.** Неверное сравнение. Нестрогое сравнение приводит к тому, что новое число, последняя цифра которого равна текущей максимальной, вытесняет текущее число, даже если новое число меньше.

**Вторая ошибка.** Неверное присваивание. При равенстве последних цифр фиксируется не введённое число, а его последняя цифра. Из-за первой

ошибки эта ошибка не проявляется (соответствующее условие никогда не выполняется), но после исправления первой ошибки возможно получение неверного результата.

### **Пример исправления для алгоритмического языка**

#### ***Первая ошибка:***

Исходная строка:

```
если dx >= mod(m,10)
```

Исправленная строка:

```
если dx > mod(m,10)
```

#### ***Вторая ошибка.***

Исходная строка:

```
то m := dx
```

Исправленная строка:

```
то m := x
```

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

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. Привести пример данных, при вводе которых программа выводит верный ответ.</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 заданных входных значениях.</p> <p>Действие 2 считается выполненным, если приведён пример данных, при вводе которых программа выводит верный ответ.</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
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций	2

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	
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 2020 положительных целых чисел, не превышающих 15 000. Необходимо найти минимальный элемент, двоичная запись которого заканчивается не более чем двумя нулями, уменьшить все чётные элементы массива, превышающие найденный минимум, на величину этого минимума и вывести изменённый массив. Если в массиве нет элементов, двоичная запись которых заканчивается не более чем двумя нулями, нужно вывести массив без изменений.

Например, для массива из шести элементов, равных 12, 13, 8, 19, 10, 14, нужно получить и вывести массив, содержащий числа 2, 13, 8, 19, 10, 4.

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=2020 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 = 2020 for i in range(0, N):     a.append(int(input())) ...</pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre>алг нач     цел N=2020     целтаб a[1:N]     цел i, m, k     нц для i от 1 до N         ввод a[i]     кц     ... кон</pre>	<pre>const     N=2020; 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>	

```
#include <iostream>
using namespace std;
const int N=2020;
int main(){
    int a[N];
    int i, m, k;
    for (i=0; i<N; ++i)
        cin >> a[i];

    ...
    return 0;
}
```

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.

<p align="center"><b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Задача решается в два прохода: на первом проходе определяется минимум, 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> <p>Числа, двоичная запись которых заканчивается тремя и более нулями, кратны 8, значит, требуется искать минимум среди чисел, не кратных 8</p>
<p><b>Пример правильной программы на алгоритмическом языке</b></p>
<pre>m := 15001 нц для i от 1 до N     если mod(a[i],8)&gt;0 и a[i]&lt;m         то m := a[i]     все кц нц для i от 1 до N     если mod(a[i],2)=0 и a[i]&gt;m         то a[i] := a[i] - m     все     вывод a[i], ' ' кц</pre>
<p>Вместо сравнения "&gt;0" можно использовать сравнение на неравенство: "&lt;&gt;0".</p> <p>В приведённом решении нет необходимости отдельно рассматривать случай, когда в массиве нет элементов, не кратных 8. В этом случае <i>m</i> сохранит значение 15 001, на втором проходе условие <i>a[i]&gt;m</i> не будет выполнено ни для одного элемента, массив будет выведен без изменений.</p> <p>При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка</p>

**Пример правильной программы на языке Python**

```
m = min((k for k in a if k%8 > 0), default = 15001)
for i in range(0,N):
    if a[i]%2 == 0 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 алгоритмическом языке

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>	
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> <ol style="list-style-type: none"> <li>1) Отсутствие инициализации или неверная инициализация минимума.</li> <li>2) Неверное определение количества нулей в конце двоичной записи числа.</li> <li>3) Неверное построение логических условий (неверные логические операции, проверка не всех условий).</li> <li>4) Выход за границы массива.</li> <li>5) Исходный массив не изменяется.</li> <li>6) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться</li> </ol>	1

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 один камень** в одну из куч и **два камня** в другую или же **увеличить** количество камней **в любой куче в два раза**. Например, пусть в одной куче 6 камней, а в другой 8 камней; такую позицию мы будем обозначать (6, 8). За один ход из позиции (6, 8) можно получить любую из четырёх позиций: (7, 10), (8, 9), (12, 8), (6, 16). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 41 или больше камней.

В начальный момент в первой куче было 8 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 32$ .

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.

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

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

### Задание 2.

Укажите максимальное значение  $S$ , при котором Петя не может выиграть первым ходом, но у Пети есть выигрышная стратегия, следуя которой, он

может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанного значения  $S$  опишите выигрышную стратегию Пети.

### Задание 3.

Проанализируйте игру при  $S = 11$ . У кого из игроков в этом случае есть выигрышная стратегия? Опишите эту стратегию и постройте дерево всех партий, возможных при этой выигрышной стратегии, 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  $S = 17, \dots, 32$ . Для выигрыша достаточно удвоить количество камней во второй куче. При меньших значениях  $S$  за один ход нельзя получить 41 или более камней в двух кучах.

б) Такая ситуация возможна при  $S = 9$ . Если Петя удвоит первую кучу, получится позиция (16, 9), из которой Ваня может получить позицию (32, 9) и выиграть. При  $S < 9$  никакой первый ход Пети не создаст ситуацию, в которой Ваня может сразу выиграть.

#### Задание 2.

Петя не может выиграть первым ходом, но может выиграть вторым ходом при  $S = 14$ . Для победы Пете нужно сделать ход (10, 15). После хода Вани возникнет одна из позиций (11, 17), (12, 16), (20, 15), (10, 30). В любой из перечисленных позиций Петя может выиграть, удвоив количество камней во второй куче.

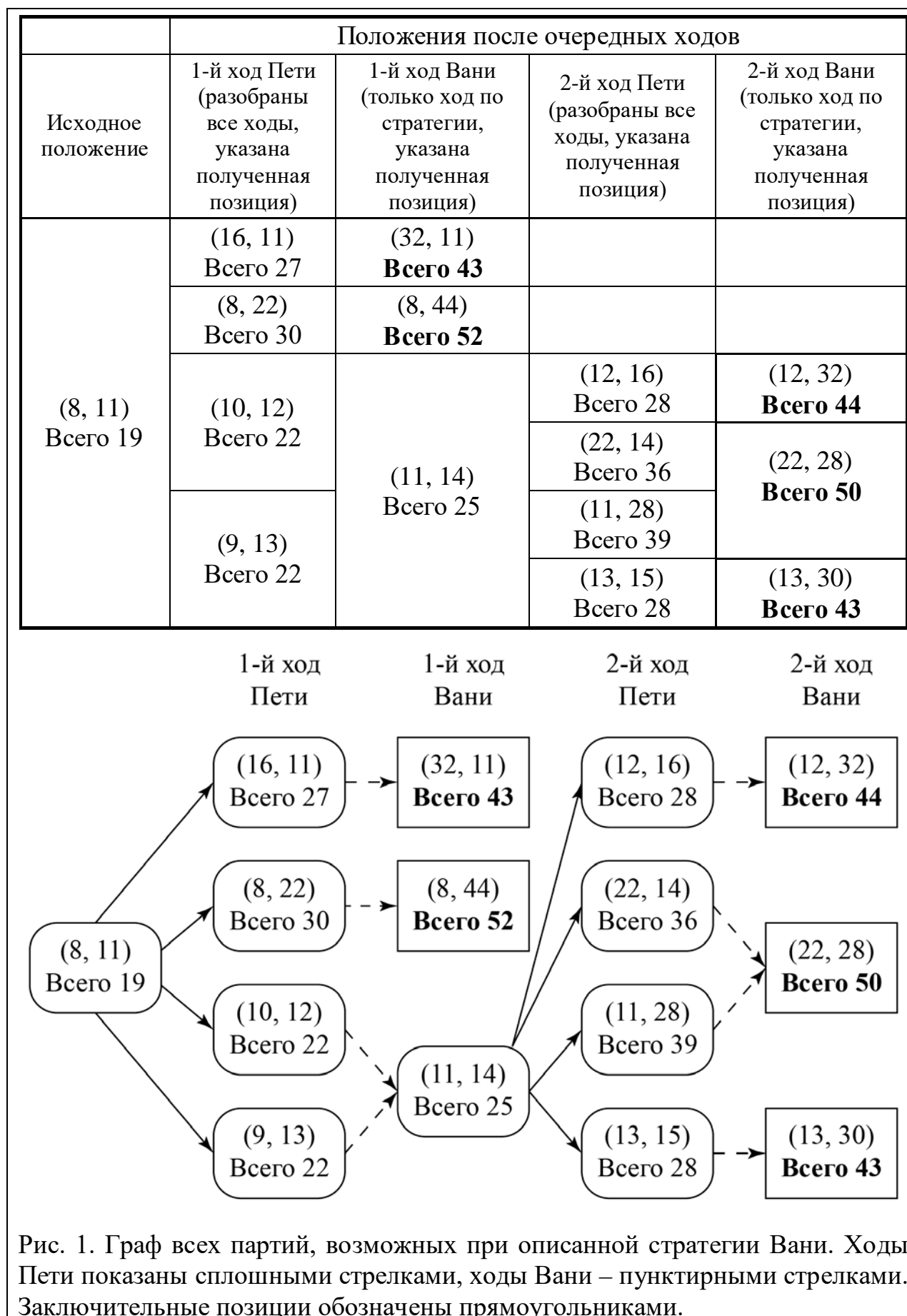
Значение  $S = 14$  максимально, так как при  $S = 15$  и  $S = 16$  после любого хода Пети Ваня может выиграть следующим ходом, а при  $S \geq 17$  Петя может выиграть первым ходом.

#### Задание 3.

Выигрышная стратегия есть у Вани. После первого хода Пети возможны позиции (16, 11), (8, 22), (10, 12), (9, 13). В позициях (16, 11) и (8, 22) Ваня может выиграть первым ходом, удвоив количество камней в большей куче. Из позиций (10, 12) и (9, 13) Ваня может получить позицию (11, 14). После второго хода Пети получится одна из позиций (22, 14), (11, 28), (12, 16), (13, 15), в любой из них Ваня может удвоить количество камней в большей куче и выиграть.

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)





*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 указано значение <math>S</math> и описана соответствующая стратегия.</p> <p>Задание 3 выполнено, если правильно указан победитель (Ваня) 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. Необходимо определить количество пар элементов этой последовательности, сумма которых делится на  $m = 80$  и при этом хотя бы один элемент из пары больше  $b = 50$ .

**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:*

6  
40  
40  
120  
30  
50  
110

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

3

*Пояснение.* Из данных шести чисел можно составить три пары, удовлетворяющие условию: (40, 120), (40, 120), (50, 110). У пар (40, 40) и (30, 50) сумма делится на 80, но оба элемента в этих парах не превышают 50.

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 элементов кратна <math>m</math>, если сумма их остатков от деления на <math>m</math> равна <math>m</math>.</p> <p>Создадим два массива по <math>m</math> элементов в каждом и будем хранить в них количество элементов последовательности, имеющих соответствующий остаток от деления на <math>m</math>; в массиве <math>a0</math> будем подсчитывать элементы, не превышающие <math>b</math>, в массиве <math>a1</math> – превышающие.</p> <p>После завершения ввода количество подходящих пар с меньшим остатком <math>p</math> от 1 до 39 можно подсчитать по формуле</p> $(a0[p]+a1[p])*a1[m-p] + a1[p]*a0[m-p]$ <p>Для остатков 0 и 40 остаток у чисел из пары совпадает, поэтому количество пар для этих остатков равно</p> $a0[p]*a1[p] + a1[p]*(a1[p]-1)/2$ <p>Общее количество пар можно найти как сумму пар по всем остаткам.</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 = 80, b=50     цел таб a0[0:m-1], a1[0:m-1]     цел N     цел x   очередное число из последовательности     цел p   остаток     цел s   количество пар     цел i      нц для i от 0 до m-1         a0[i] := 0; a1[i] := 0     кц     ввод N     нц N раз         ввод x         p:=mod(x,m)         если x &lt;= b             то a0[p] := a0[p]+1             иначе a1[p] := a1[p]+1         все     кц     p := 0     s := a0[p]*a1[p] + div(a1[p]*(a1[p]-1),2)     p := div(m,2)     s := s + a0[p]*a1[p] + div(a1[p]*(a1[p]-1),2) </pre>

```

нц для p от 1 до div(m,2)-1
  s := s + (a0[p]+a1[p])*a1[m-p] + a1[p]*a0[m-p]
кц
ВЫВОД s
кОН

```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и подсчитаем подходящие. 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;
      b = 50;

var
  N: integer;      {количество чисел}
  a: array [1..10000] of integer; {исходные данные}
  s: integer;      {ответ - количество пар}
  i, j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  s:=0;
  for i := 1 to N-1 do begin
    for j := i+1 to N do begin
      if ((a[i] + a[j]) mod m = 0) and
          ((a[i] > b) or (a[j]) > b))
      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	
Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел $N$ , время работы пропорционально $N$ и $m$ . Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов: 1) пропущен или неверно указан знак пунктуации;	4

<p>2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования;</p> <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> и параметру <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 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, в том числе для логических операций;</li> <li>6) отсутствует вывод ответа или выводится не то значение;</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>	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>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. В частности, в 2 балла оценивается программа, в которой верно подсчитывается количество элементов по остаткам, но допущены ошибки при подсчёте количества пар, например, пары подсчитаны дважды (неверные границы заключительного цикла перебора остатков), или неверно обработаны особые значения остатков</p>	2

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, рассмотрение пар 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) подсчитывается количество подходящих пар	1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл</i>	<i>4</i>

**Критерии оценивания заданий с развёрнутым ответом****24**

Даны 4 целых положительных числа. Необходимо выбрать из них и вывести на экран число с наибольшей последней цифрой. Если в наборе несколько чисел с одинаковой наибольшей последней цифрой, нужно вывести наибольшее из этих чисел.

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 M, X, DX AS INTEGER M = 1 FOR I = 1 to 4     INPUT X     DX = X MOD 10     IF DX &gt; M MOD 10 THEN         M = X     ELSE         IF DX=M MOD 10 OR X&gt;M THEN             M = X         END IF     END IF NEXT I PRINT M </pre>	<pre> m = 1 for i in range (4):     x = int(input())     dx = x % 10     if dx &gt; m%10:         m = x     else:         if dx == m%10 or x&gt;m:             m = x print(m) </pre>
<b>C++</b>	<b>Паскаль</b>
<pre> #include &lt;iostream&gt; using namespace std; int main() {     int m, x, dx, i;     m = 1;     for (i=1; i&lt;=4; ++i) {         cin &gt;&gt; x;         dx = x%10;         if (dx &gt; m%10)             m = x;         else {             if (dx==m%10    x&gt;m)                 m = x;         }     }     cout &lt;&lt; m;     return 0; } </pre>	<pre> var m, x, dx, i: integer; begin     m := 1;     for i:=1 to 4 do begin         readln(x);         dx := x mod 10;         if dx &gt; m mod 10             then m :=x         else begin             if (dx=m mod 10) or (x&gt;m)                 then m :=x             end         end;         write(m)     end. end. </pre>



**Алгоритмический язык**

```

алг
нач
  цел m, x, dx, i
  m := 1
  для i от 1 до 4
    ввод x
    dx := mod(x,10)
    если dx > mod(m,10)
      то m := x
    иначе
      если dx = mod(m,10) или x>m
        то m := x
      все
    все
  кц
  вывод m
кон

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 19, 29, 104, 16.
2. Приведите пример исходных данных, при вводе которых программа, несмотря на ошибки, 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)

1. При вводе указанных чисел программа выведет число 16.
2. Программа выводит верный ответ А, если после ввода А не вводится никакое число, которое больше А. Пример такого набора: 15, 20, 37, 32.
3. Программа содержит две ошибки.

**Первая ошибка.** Неверная инициализация. Если все числа в наборе заканчиваются цифрой 0, то (в случае исправления второй ошибки) вместо правильного ответа будет выводиться 1.

**Вторая ошибка.** Неверное логическое условие. При рассмотрении случая, когда последняя цифра нового числа равна текущей максимальной,

используется неверная логическая операция. В результате, если новое число больше числа с максимальной последней цифрой, будет зафиксировано новое число, даже если его последняя цифра меньше.

**Пример исправления для алгоритмического языка**

**Первая ошибка:**

Исходная строка:

`m := 1`

Исправленная строка:

`m := 0`

**Вторая ошибка.**

Исходная строка:

`если dx = mod(m, 10) или x > m`

Исправленная строка:

`если dx = mod(m, 10) и x > 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

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. Привести пример данных, при вводе которых программа выводит верный ответ.</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 заданных входных значениях.</p> <p>Действие 2 считается выполненным, если приведён пример данных, при вводе которых программа выводит верный ответ.</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>	
<p>Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций</p>	2

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	
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 2020 положительных целых чисел, не превышающих 15 000. Необходимо найти минимальный элемент, двоичная запись которого заканчивается не более чем одним нулём, уменьшить все нечётные элементы массива, превышающие найденный минимум, на величину этого минимума и вывести изменённый массив. Если в массиве нет элементов, двоичная запись которых заканчивается не более чем одним нулём, нужно вывести массив без изменений.

Например, для массива из шести элементов, равных 12, 13, 18, 19, 20, 15, нужно получить и вывести массив, содержащий числа 12, 13, 18, 6, 20, 2.

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=2020 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 = 2020 for i in range(0, N):     a.append(int(input())) ...</pre>
<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre>алг нач     цел N=2020     целтаб a[1:N]     цел i, m, k     нц для i от 1 до N         ввод a[i]     кц     ... кон</pre>	<pre>const     N=2020; var     a: array [1..N] of integer;     i, m, k: integer; begin     for i:=1 to N do         readln(a[i]);     ... end.</pre>

**C++**

```
#include <iostream>
using namespace std;
const int N=2020;
int main(){
    int a[N];
    int i, m, k;
    for (i=0; i<N; ++i)
        cin >> a[i];

    ...
    return 0;
}
```

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 определяется минимум, 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.

Числа, двоичная запись которых заканчивается двумя и более нулями, кратны 4, значит, требуется искать минимум среди чисел, не кратных 4

**Пример правильной программы на алгоритмическом языке**

```
m := 15001
нц для i от 1 до N
    если mod(a[i],4)>0 и a[i]<m
        то m := a[i]
    все
кц
нц для i от 1 до N
    если mod(a[i],2)=1 и a[i]>m
        то a[i] := a[i] - m
    все
    вывод a[i], ' '
кц
```

Вместо сравнения ">0" можно использовать сравнение на неравенство: "<>0".

Вместо сравнения "=1" можно использовать неравенство ">0" или "<>0".

В приведённом решении нет необходимости отдельно рассматривать случай, когда в массиве нет элементов, не кратных 4. В этом случае *m* сохранит значение 15 001, на втором проходе условие  $a[i] > m$  не будет выполнено ни для одного элемента, массив будет выведен без изменений

При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка

**Пример правильной программы на языке Python**

```
m = min((k for k in a if k%4 > 0), default = 15001)
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 алгоритмическом языке

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>	
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> <ol style="list-style-type: none"> <li>1) Отсутствие инициализации или неверная инициализация минимума.</li> <li>2) Неверное определение количества нулей в конце двоичной записи числа.</li> <li>3) Неверное построение логических условий (неверные логические операции, проверка не всех условий).</li> <li>4) Выход за границы массива.</li> <li>5) Исходный массив не изменяется.</li> <li>6) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться</li> </ol>	1

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 один камень** в одну из куч и **два камня** в другую или же **увеличить** количество камней **в любой куче в два раза**. Например, пусть в одной куче 6 камней, а в другой 8 камней; такую позицию мы будем обозначать (6, 8). За один ход из позиции (6, 8) можно получить любую из четырёх позиций: (7, 10), (8, 9), (12, 8), (6, 16). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

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

В начальный момент в первой куче было 10 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 36$ .

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.

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

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

**Задание 2.**

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

**Задание 3.**

Проанализируйте игру при  $S = 13$ . У кого из игроков в этом случае есть выигрышная стратегия? Опишите эту стратегию и постройте дерево всех партий, возможных при этой выигрышной стратегии, 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  $S = 19, \dots, 36$ . Для выигрыша достаточно удвоить количество камней во второй куче. При меньших значениях  $S$  за один ход нельзя получить 47 или более камней в двух кучах.
- б) Такая ситуация возможна при  $S = 7$ . Если Петя удвоит первую кучу, получится позиция  $(20, 7)$ , из которой Ваня может получить позицию  $(40, 7)$  и выиграть. При  $S < 7$  никакой первый ход Пети не создаст ситуацию, в которой Ваня может сразу выиграть.

**Задание 2.**

Петя не может выиграть первым ходом, но может выиграть вторым ходом при  $S = 16$ . Для победы Пете нужно сделать ход  $(12, 17)$ . После хода Вани возникнет одна из позиций  $(14, 18)$ ,  $(13, 19)$ ,  $(24, 17)$ ,  $(12, 34)$ . В любой из перечисленных позиций Петя может выиграть, удвоив количество камней во второй куче.

Значение  $S = 16$  максимально, так как при  $S = 17$  и  $S = 18$  после любого хода Пети Ваня может выиграть следующим ходом, а при  $S \geq 19$  Петя может выиграть первым ходом.

**Задание 3.**

Выигрышная стратегия есть у Вани. После первого хода Пети возможны позиции  $(20, 13)$ ,  $(10, 26)$ ,  $(12, 14)$ ,  $(11, 15)$ . В позициях  $(20, 13)$  и  $(10, 26)$  Ваня может выиграть первым ходом, удвоив количество камней в большей куче. Из позиций  $(12, 14)$  и  $(11, 15)$  Ваня может получить позицию  $(13, 16)$ . После второго хода Пети получится одна из позиций  $(26, 16)$ ,  $(13, 32)$ ,  $(14, 18)$ ,  $(15, 17)$ , в любой из них Ваня может удвоить количество камней в большей куче и выиграть.

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, 13) Всего 23	(20, 13) Всего 33	(40, 13) <b>Всего 53</b>		
	(10, 26) Всего 36	(10, 52) <b>Всего 62</b>		
	(11, 15) Всего 26	(13, 16) Всего 29	(14, 18) Всего 32	(14, 36) <b>Всего 50</b>
	(12, 14) Всего 26		(13, 32) Всего 45	(26, 32) <b>Всего 58</b>
			(26, 16) Всего 42	
				(15, 17) Всего 32

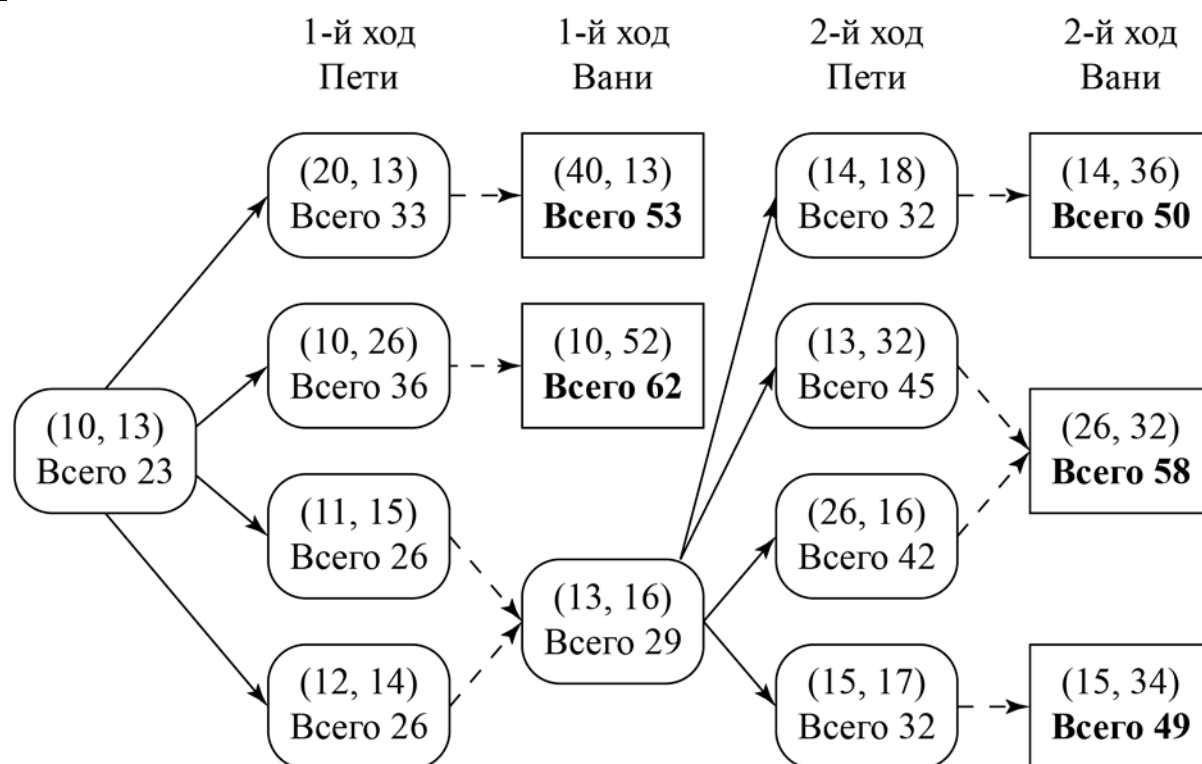


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*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 указано значение <math>S</math> и описана соответствующая стратегия.</p> <p>Задание 3 выполнено, если правильно указан победитель (Ваня) 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. Необходимо определить количество пар элементов этой последовательности, сумма которых делится на  $m = 60$  и при этом хотя бы один элемент из пары больше  $b = 40$ .

**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:*

6  
30  
30  
150  
20  
40  
100

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

3

*Пояснение.* Из данных шести чисел можно составить три пары, удовлетворяющие условию: (30, 150), (30, 150), (20, 100). У пар (30, 30) и (20, 40) сумма делится на 60, но оба элемента в этих парах не превышают 40.

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 элементов кратна  $m$ , если сумма их остатков от деления на  $m$  равна  $m$ .

Создадим два массива по  $m$  элементов в каждом и будем хранить в них количество элементов последовательности, имеющих соответствующий остаток от деления на  $m$ ; в массиве  $a_0$  будем подсчитывать элементы, не превышающие  $b$ , в массиве  $a_1$  – превышающие.

После завершения ввода количество подходящих пар с меньшим остатком  $p$  от 1 до 29 можно подсчитать по формуле

$$(a_0[p] + a_1[p]) * a_1[m-p] + a_1[p] * a_0[m-p]$$

Для остатков 0 и 30 остаток у чисел из пары совпадает, поэтому количество пар для этих остатков равно

$$a_0[p] * a_1[p] + a_1[p] * (a_1[p] - 1) / 2$$

Общее количество пар можно найти как сумму пар по всем остаткам.

Ниже приведена программа на алгоритмическом языке, реализующая этот алгоритм

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

алг задача27

нач

цел  $m = 60, b = 40$

цел таб  $a_0[0:m-1], a_1[0:m-1]$

цел  $N$

цел  $x$  | очередное число из последовательности

цел  $p$  | остаток

цел  $s$  | количество пар

цел  $i$

нц для  $i$  от 0 до  $m-1$

$a_0[i] := 0; a_1[i] := 0$

кц

ввод  $N$

нц  $N$  раз

ввод  $x$

$p := \text{mod}(x, m)$

если  $x \leq b$

то  $a_0[p] := a_0[p] + 1$

иначе  $a_1[p] := a_1[p] + 1$

все

кц

$p := 0$

$s := a_0[p] * a_1[p] + \text{div}(a_1[p] * (a_1[p] - 1), 2)$

$p := \text{div}(m, 2)$

```

s := s + a0[p]*a1[p] + div(a1[p]*(a1[p]-1),2)
нц для p от 1 до div(m,2)-1
    s := s + (a0[p]+a1[p])*a1[m-p] + a1[p]*a0[m-p]
кц
вывод s
кон

```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и подсчитаем подходящие. 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;
      b = 40;
var
  N: integer;      {количество чисел}
  a: array [1..10000] of integer; {исходные данные}
  s: integer;      {ответ - количество пар}
  i,j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  s:=0;
  for i := 1 to N-1 do begin
    for j := i+1 to N do begin
      if ((a[i] + a[j]) mod m = 0) and
        ((a[i] > b) or (a[j]) > b))
      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	
Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел $N$ , время работы пропорционально $N$ и $m$ .	4

<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>	
<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> и параметру <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, в том числе для логических операций;</li> <li>6) отсутствует вывод ответа или выводится не то значение;</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>	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>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. В частности, в 2 балла оценивается программа, в которой верно подсчитывается количество</p>	2

<p>элементов по остаткам, но допущены ошибки при подсчёте количества пар, например, пары подсчитаны дважды (неверные границы заключительного цикла перебора остатков), или неверно обработаны особые значения остатков.</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, рассмотрение пар вида <math>a[i]+a[i]</math> и т. д.</p>	
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.</p> <p>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) подсчитывается количество подходящих пар</li> </ol>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<i>Максимальный балл</i>	4