

Критерии оценивания заданий с развёрнутым ответом**24**

Факториалом натурального числа n (обозначается $n!$) называется произведение всех натуральных чисел от 1 до n . Например, $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$.

Дано целое положительное число A . Необходимо найти **минимальное** натуральное K , для которого $K! > A$. 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 A, K, F AS INTEGER INPUT A K = 2 F = 1 WHILE F < A K = K + 1 F = F * K WEND PRINT K END </pre>	<pre> a = int(input()) k = 2 f = 1 while f < a: k += 1 f *= k print(k) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел a, k, f ввод a k := 2 f := 1 нц пока f < a k := k + 1 f := f * k кц вывод k кон </pre>	<pre> var a, k, f: integer; begin read(a); k := 2; f := 1; while f < a do begin k := k + 1; f := f * k end; writeln(k) end. </pre>
C++	
<pre> #include <iostream> using namespace std; int main(){ int a, k, f; cin >> a; k = 2; f = 1; while (f < a) { ++k; f *= k; } cout << k; return 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе $A = 5$.
2. Назовите *минимальное A , большее 10*, при котором программа выведет *неверный* ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

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 $A = 5$ программа выведет число 4. 2. Минимальное $A > 10$, при котором программа выдает <i>неверный</i> ответ, равно 13. Программа выдаёт ответ 5, правильный ответ равен 4. При вводе меньших чисел, 11 и 12, программа выдаёт верный ответ 4. 3. Программа содержит две ошибки. 1) Неверная инициализация. Значения переменных F и K не согласованы: эти переменные могут быть равны 1 или 2, но они должны быть одинаковы. 2) Неверное условие цикла.	
Пример исправления для языка Паскаль: Первая ошибка, способ 1: $k := 2;$ Исправленная строка: $k := 1;$ Первая ошибка, способ 2: $f := 1;$ Исправленная строка: $f := 2;$ Вторая ошибка: while $f < a$ do begin Исправленная строка: while $f \leq a$ do begin	
В программах на других языках ошибочные строки и их исправления аналогичны. Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления	
Указания по оцениванию	Баллы
В задаче требуется выполнить три действия.	
1. Указать ответ программы при данном вводе.	

Указания по оцениванию	Баллы
<p>Это действие считается выполненным, если указан верный результат работы программы при заданном входном значении. Учащийся не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать исходные данные, соответствующие требованиям. Это действие считается выполненным, если указано верное значение исходного числа. Учащийся не обязан пояснять, как получено это число и доказывать, что оно удовлетворяет требованиям.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе ошибки и предложены верные варианты исправления, при этом никакие верные строки программы не указаны в качестве неверных. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка). Ошибка считается исправленной, если выполнены оба следующих условия:</p> <ul style="list-style-type: none"> а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении ошибки получается правильная программа 	
Выполнены все три необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <ol style="list-style-type: none"> 1. Выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной. 2. Выполнены два первых действия, найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной. 3. Выполнено одно из первых двух действий, найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной 	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев:</p> <ol style="list-style-type: none"> 1. Выполнены два первых действия. При этом несущественно, насколько правильно выполнено третье действие 2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2. 3. Выполнено одно из двух первых действий. Исправлена одна из двух ошибок. Не более чем одна верная строка названа ошибочной 	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

25

Дан массив, содержащий 2018 положительных целых чисел, не превышающих 30 000. Необходимо определить, сколько в этом массиве элементов, десятичная и восьмеричная запись которых содержит одинаковое количество цифр.

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=2018 DIM A(N) AS INTEGER DIM B, I, K, L, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre># кроме уже указанных # допускается использование # целочисленных переменных # k, b, l, m a = [] N = 2018 for i in range(0, N): a.append(int(input())) ...</pre>
Алгоритмический язык	Паскаль
<pre>алг нач цел N=2018 целтаб a[1:N] цел b, i, k, l, m нц для i от 1 до N ввод a[i] кц ... кон</pre>	<pre>const N=2018; var a: array [1..N] of integer; b, i, k, l, m: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>
C++	
<pre>#include <iostream> using namespace std; const int N=2018; int main(){ int a[N]; int b, i, k, l, m; for (i=0; i<N; ++i) cin >> 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.

<p align="center">Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Необходимо перебрать все элементы массива, для каждого из них определить, верно ли, что данное число записывается одинаковым количеством цифр в десятичной и восьмеричной системах, и подсчитать количество элементов, для которых это условие выполняется. Определить совпадение количества цифр можно разными способами.</p> <p>Способ 1. Количество цифр в записи числа можно определить с помощью последовательного деления числа на основание системы счисления</p>
<p align="center">Пример правильной программы на языке Паскаль</p>
<pre> k := 0; for i:=1 to N do begin l := 0; b := a[i]; while b > 0 do begin l := l+1; b := b div 8 end; m := 0; b := a[i]; while b > 0 do begin m := m+1; b := b div 10 end; if l = m then k := k+1 end; writeln(k) </pre>
<p>Способ 2. Количество цифр в записи числа можно определить с помощью проверки на принадлежность к заранее заданному диапазону</p>
<p align="center">Пример правильной программы на алгоритмическом языке</p>
<pre> k := 0 нц для i от 1 до N выбор при 1 <= a[i] <= 7: l := 1 при 8 <= a[i] <= 63: l := 2 при 64 <= a[i] <= 511: l := 3 при 512 <= a[i] <= 4095: l := 4 при 4096 <= a[i] : l := 5 все выбор при 1 <= a[i] <= 9: m := 1 при 10 <= a[i] <= 99: m := 2 при 100 <= a[i] <= 999: m := 3 при 1000 <= a[i] <= 9999: m := 4 при 10000 <= a[i] : m := 5 все если l=m то k := k+1 все кц </pre>

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)	
Вывод k	
Способ 3. Можно не определять количество цифр, а заранее найти диапазоны значений, в которых количество цифр в восьмеричной и десятичной записях совпадает	
Пример правильной программы на языке C++	
<pre> k = 0; for (i=0; i<N; ++i) { if (1 <= a[i] && a[i] <= 7 10 <= a[i] && a[i] <= 63 100 <= a[i] && a[i] <= 511 1000 <= a[i] && a[i] <= 4095 10000 <= a[i]) ++k; } cout << k; </pre>	
Способ 4. Этот способ можно использовать в тех языках программирования, где есть стандартные средства, позволяющие получить строковые представления целого числа в восьмеричной и десятичной системах. Например, в языке Python это можно сделать с помощью функций oct и str. В приведённой ниже программе учтено, что функция oct в Python 3 добавляет к восьмеричной записи числа двухсимвольный префикс 0o, но неучёт этого префикса в работе учащегося не считается ошибкой	
Пример правильной программы на языке Python	
<pre> k = 0 for b in a: if len(oct(b)[2:]) == len(str(b)): k += 1 print(k) </pre>	
Язык Python позволяет реализовать описанный выше способ с помощью одной строки: <pre>print(sum(1 for b in a if len(oct(b)[2:]) == len(str(b))))</pre>	

Указания по оцениванию	Баллы
Предложена правильная программа, выдающая верный ответ для любых соответствующих условию исходных данных. Допускается запись программы на языке, не входящем в список языков из условия. Такая программа должна использовать переменные, аналогичные заданным 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	2

Указания по оцениванию	Баллы
<p>только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи.</p> <p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора</p>	
<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"> 1) Отсутствие инициализации или неверная инициализация счётчика совпадений. 2) Выход за границы массива. 3) Изменение исходных данных в процессе решения (например, определение количества цифр с помощью деления на основание системы счисления самого элемента, а не его копии). 4) Неверная инициализация и/или построение цикла при определении количества цифр с помощью деления. Если при этом в построении обоих циклов допущена ошибка, которая всегда приводит к одинаковому изменению верного количества (например, счётчик количества цифр инициализируется единицей вместо нуля, при этом получится ответ на 1 больше правильного), то это искажение не влияет на окончательный ответ и, следовательно, не считается ошибкой. 5) Неверное определение одной или нескольких границ диапазона (с учётом использованных знаков сравнения) при решении с использованием диапазонов. 6) Неверное построение логического выражения (неверная расстановка скобок с учётом приоритетов действий в конкретном языке программирования, неверное использование операций И и ИЛИ). 7) Отсутствует вывод ответа. 8) Используется переменная, не объявленная в разделе описания переменных. 9) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно 	1
Не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может

увеличить количество камней в куче в два раза или

увеличить количество камней в куче в три раза.

Например, имея кучу из 10 камней, за один ход можно получить кучу из 20 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче превышает 49. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 50 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 49$.

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 следующей стратегии игрока, которые не являются для него безусловно выигрышными.

Выполните следующие задания.

Задание 1. Назовите все значения S , при которых Петя может выиграть первым ходом, причём у Пети есть **ровно один** выигрышающий ход.

Задание 2. Назовите все значения S , при которых Ваня может выиграть первым ходом, независимо от того, каким будет первый ход Пети. Опишите выигрышную стратегию Вани для этих значений S .

Задание 3. Назовите все значения S , при которых Петя не может выиграть первым ходом, но может выиграть вторым ходом независимо от того, как будет играть Ваня, причём в начальной позиции у Пети есть **ровно один** выигрышающий ход. Опишите выигрышную стратегию Пети для всех этих значений. Постройте (в виде рисунка или таблицы) дерево всех партий, возможных при этой стратегии для одного произвольного значения S . На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции. Дерево должно содержать только те партии, которые возможны при реализации выигрышной стратегии Пети.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание 1. Петя может выиграть единственным первым ходом, если $S = 17, \dots, 24$. Для выигрыша необходимо утроить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой будет 50 или более камней. При больших значениях S Петя тоже выигрывает первым ходом, но сделать это можно двумя способами (удвоением и утроением)

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание 2. Ваня выигрывает первым ходом при S от 9 до 16. После хода Пети в куче станет от 18 до 48 камней. В любом из этих случаев Ваня может утроить количество камней и выиграть.

Задание 3. Чтобы выиграть вторым ходом, Петя должен сделать так, чтобы после его первого хода в куче было от 9 до 16 камней. Эти позиции разобраны в задании 2, в них игрок, который будет ходить (теперь это Ваня), проигрывает. Петя может получить такую кучу единственным способом при $S = 3, 4, 6, 7, 8$. При $S = 3$ и $S = 4$ Петя должен первым ходом утроить количество камней, в остальных случаях – удвоить. После этого в куче будет от 9 до 16 камней, после ответного хода Вани Петя может утроить количество камней и выиграть. Значение $S = 5$ не входит в список, так как в этом случае у Пети есть два выигрывающих хода.

В таблице изображено дерево возможных партий при описанной стратегии Пети для $S = 6$. Заключительные позиции (в них выигрывает Петя) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы). Для других значений S дерево строится аналогично.

Исходное положение	1-й ход Пети (только ход по стратегии)	1-й ход Вани (разобраны все ходы)	2-й ход Пети (только ход по стратегии)
6	$6*2 = \mathbf{12}$	$12*2 = \mathbf{24}$	$24*3 = \underline{\mathbf{72}}$
		$12*3 = \mathbf{36}$	$36*3 = \underline{\mathbf{108}}$

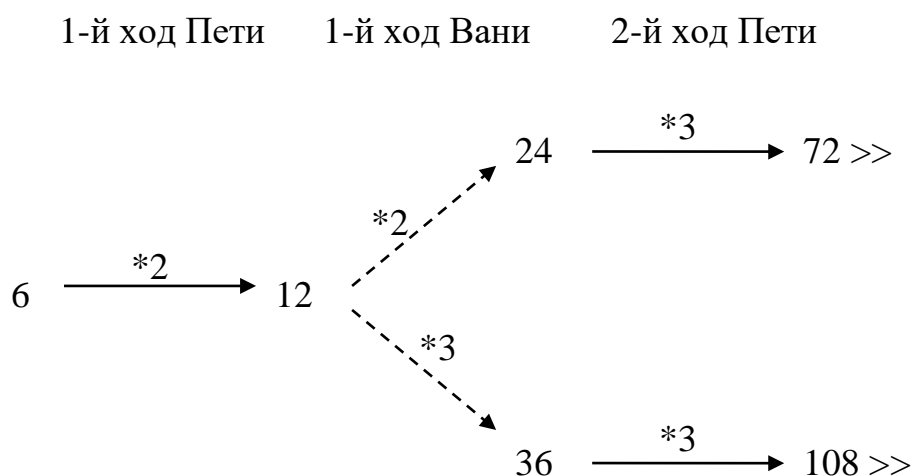


Рис. 1. Дерево всех партий, возможных при описанной стратегии Пети. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены знаком $>>$.

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить три задания. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже). Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается. Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены все три задания. Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. 1. Выполнено задание 3. 2. Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий. 1. Выполнено задание 1. 2. Выполнено задание 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<i>3</i>

27

Дан набор из N целых положительных чисел. Из этих чисел формируются все возможные пары (парой считаются два элемента, которые находятся на разных местах в наборе, порядок чисел в паре не учитывается), в каждой паре вычисляется сумма элементов. Необходимо определить количество пар, для которых полученная сумма делится на 7.

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

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

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 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.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$). В каждой из последующих N строк записано одно натуральное число, не превышающее 10 000.

Пример входных данных:

5
1
3
6
11
1

Пример выходных данных для приведенного выше примера входных данных:

3

Из 5 чисел можно составить 10 пар. В данном случае у трёх пар сумма делится на 7: $1 + 6$, $1 + 6$ (в наборе две единицы, поэтому пару $1 + 6$ можно составить двумя способами), $3 + 11$.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Разобьём все числа исходного набора на 7 групп по значению остатка от деления на 7 и подсчитаем количество чисел в каждой группе. Сами числа можно не хранить, достаточно при вводе определить остаток от деления очередного числа на 7 и увеличить соответствующий счётчик. Таким образом, независимо от количества чисел в исходном наборе, после чтения входных данных для хранения необходимой информации хватит массива из 7 элементов и программа получится эффективной по памяти.</p> <p>Чтобы сумма двух чисел делилась на 7, они оба должны делиться на 7 либо сумма их остатков от деления на 7 должна быть равна 7. Зная количество элементов для каждого остатка, можно определить количество пар.</p> <p>Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)</p>

Пример правильной и эффективной программы на языке Паскаль

```

var
  N: integer;      {количество чисел}
  a: integer;      {очередное число}
  d: array [0..6] of integer; {группы по остаткам}
  s: integer;      {количество пар}
  i: integer;

begin
  for i:=0 to 6 do d[i]:=0;
  readln(N);
  for i:=1 to N do begin
    readln(a);
    inc(d[a mod 7])
  end;
  s := d[0]*(d[0]-1) div 2;
  for i:=1 to 3 do s := s + d[i]*d[7-i];
  writeln(s)
end.

```

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

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

Пример правильной, но неэффективной программы на языке Паскаль

```

var
  N: integer;      {количество чисел}
  a: array [1..1000] of integer; {исходные данные}
  s: integer;      {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 7 = 0
      then s := s + 1
    end
  end;
  writeln(s)
end.

```

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> 1) пропущен или неверно указан знак пунктуации; 2) неверно написано или пропущено зарезервированное слово языка программирования; 3) не описана или неверно описана переменная; 4) применяется операция, недопустимая для соответствующего типа данных. <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 введенных чисел, правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в C++ или другой аналогичной структуре данных).</p> <p>Количество синтаксических ошибок («описок»), указанных в критериях на 4 балла, – не более пяти.</p> <p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> 1) ошибка при инициализации или отсутствие инициализации счётчиков; 2) использование нахождения остатка вместо деления (mod вместо div в Паскале) или наоборот; 3) допущен выход за границу массива; 4) не учтены или неверно учтены пары, в которых каждый элемент делится на 7; 5) неверно составлены или учтены не все комбинации остатков; 6) неверно подсчитано количество пар для всех или некоторых комбинаций остатков (например, не выполняется деление на 2) 	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает верно, эффективно или нет. В частности, в 2 балла оцениваются переборные решения, в которых все исходные данные сохраняются в массиве, рассматриваются все возможные пары, из которых выбираются подходящие.</p>	2

Указания по оцениванию	Баллы
Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла	
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа описывает в целом правильный алгоритм (эффективный или нет), но количество допущенных ошибок не укладывается в описанные выше ограничения	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**

Факториалом натурального числа n (обозначается $n!$) называется произведение всех натуральных чисел от 1 до n . Например, $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$.

Дано целое положительное число A . Необходимо найти **минимальное** натуральное K , для которого $K! \geq A$.

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 A, K, F AS INTEGER INPUT A K = 0 F = 1 WHILE F <= A K = K + 1 F = F * K WEND PRINT K END </pre>	<pre> a = int(input()) k = 0 f = 1 while f <= a: k += 1 f *= k print(k) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел a, k, f ввод a k := 0 f := 1 нц пока f <= a k := k + 1 f := f * k кц вывод k кон </pre>	<pre> var a, k, f: integer; begin read(a); k := 0; f := 1; while f <= a do begin k := k + 1; f := f * k end; writeln(k) end. </pre>
C++	
<pre> #include <iostream> using namespace std; int main(){ int a, k, f; cin >> a; k = 0; f = 1; while (f <= a) { ++k; f *= k; } cout << k; return 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе $A = 6$.
2. Назовите **минимальное A , большее 10**, при котором программа выведет **неверный** ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

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 $A = 6$ программа выведет число 4.
2. Минимальное $A > 10$, при котором программа выдаёт **неверный** ответ, равно 24. Программа выдаёт неверные ответы, если введённое число оказывается точным факториалом. Минимальный точный факториал, больший 10, равен 24.
3. Программа содержит две ошибки.
 - 1) Неверная инициализация. Начальное значение переменной K должно быть равно 1. Это значение не влияет на работу программы при $A > 1$, но при $A = 1$ оно оказывается существенным. Если исправить вторую ошибку, сохранив нулевое начальное значение K , то программа выдаст ответ 0 вместо правильного 1.
 - 2) Неверное условие цикла.

Пример исправления для языка Паскаль:

Первая ошибка:

```
k := 0;
```

Исправленная строка:

```
k := 1;
```

Вторая ошибка:

```
while f <= a do begin
```

Исправленная строка:

```
while f < a do begin
```

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

Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1. Указать ответ программы при данном вводе. Это действие считается выполненным, если указан верный результат работы программы при заданном входном значении. Учащийся не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать исходные данные, соответствующие требованиям. Это действие считается выполненным, если указано верное значение исходного числа. Учащийся не обязан пояснять, как получено это число и доказывать, что оно удовлетворяет требованиям.</p> <p>3. Найти и исправить ошибки в программе. Это действие считается выполненным, если верно указаны обе ошибки и предложены верные варианты исправления, при этом никакие верные строки программы не указаны в качестве неверных. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка). Ошибка считается исправленной, если выполнены оба следующих условия:</p> <ul style="list-style-type: none"> а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении ошибки получается правильная программа 	
Выполнены все три необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций.</p> <p>1. Выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.</p> <p>2. Выполнены два первых действия, найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.</p> <p>3. Выполнено одно из первых двух действий, найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной</p>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия. При этом несущественно, насколько правильно выполнено третье действие.</p> <p>2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2</p>	1

Указания по оцениванию	Баллы
3. Выполнено одно из двух первых действий. Исправлена одна из двух ошибок. Не более чем одна верная строка названа ошибочной	
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
Максимальный балл	3

25

Дан массив, содержащий 2018 положительных целых чисел, не превышающих 30 000. Необходимо определить, сколько в этом массиве элементов, десятичная и шестнадцатеричная запись которых содержит одинаковое количество цифр.

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=2018 DIM A(N) AS INTEGER DIM B, I, K, L, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre># кроме уже указанных # допускается использование # целочисленных переменных # k, b, l, m a = [] N = 2018 for i in range(0, N): a.append(int(input())) ...</pre>
Алгоритмический язык	Паскаль
<pre>алг нач цел N=2018 целтаб a[1:N] цел b, i, k, l, m нц для i от 1 до N ввод a[i] кц ... кон</pre>	<pre>const N=2018; var a: array [1..N] of integer; b, i, k, l, m: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>
C++	
<pre>#include <iostream> using namespace std; const int N=2018; int main(){ int a[N]; int b, i, k, l, m; for (i=0; i<N; ++i) cin >> 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.

<p align="center">Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Необходимо перебрать все элементы массива, для каждого из них определить, верно ли, что данное число записывается одинаковым количеством цифр в десятичной и шестнадцатеричной системах, и подсчитать количество элементов, для которых это условие выполняется. Определить совпадение количества цифр можно разными способами.</p> <p>Способ 1. Количество цифр в записи числа можно определить с помощью последовательного деления числа на основание системы счисления</p>
<p align="center">Пример правильной программы на языке Паскаль</p>
<pre> k := 0; for i:=1 to N do begin l := 0; b := a[i]; while b > 0 do begin l := l+1; b := b div 16 end; m := 0; b := a[i]; while b > 0 do begin m := m+1; b := b div 10 end; if l = m then k := k+1 end; writeln(k) </pre>
<p>Способ 2. Количество цифр в записи числа можно определить с помощью проверки на принадлежность к заранее заданному диапазону</p>
<p align="center">Пример правильной программы на алгоритмическом языке</p>
<pre> к := 0 нц для i от 1 до N выбор при 1 <= a[i] <= 15: l := 1 при 16 <= a[i] <= 255: l := 2 при 256 <= a[i] <= 4095: l := 3 при 4096 <= a[i] : l := 4 все выбор при 1 <= a[i] <= 9: m := 1 при 10 <= a[i] <= 99: m := 2 </pre>

```

    при 100 <= a[i] <= 999: m := 3
    при 1000 <= a[i] <= 9999: m := 4
    при 10000 <= a[i] : m := 5
все
если l=m
    то k := k+1
все
кц
вывод k

```

Способ 3. Можно не определять количество цифр, а заранее найти диапазоны значений, в которых количество цифр в шестнадцатеричной и десятичной записях совпадает.

Пример правильной программы на языке C++

```

k = 0;
for (i=0; i<N; ++i) {
    if (    1 <= a[i] && a[i] <=    9 ||
        16 <= a[i] && a[i] <=   99 ||
        256 <= a[i] && a[i] <=  999 ||
        4096 <= a[i] && a[i] <= 9999
    )
        ++k;
}
cout << k;

```

Способ 4. Этот способ можно использовать в тех языках программирования, где есть стандартные средства, позволяющие получить строковые представления целого числа в шестнадцатеричной и десятичной системах. Например, в языке Python это можно сделать с помощью функций hex и str. В приведённой ниже программе учтено, что функция hex в Python 3 добавляет к шестнадцатеричной записи числа двухсимвольный префикс 0x, но неучёт этого префикса в работе учащегося не считается ошибкой

Пример правильной программы на языке Python

```

k = 0
for b in a:
    if len(hex(b)[2:]) == len(str(b)):
        k += 1
print(k)

```

Язык Python позволяет реализовать описанный выше способ с помощью одной строки:

```
print(sum(1 for b in a if len(hex(b)[2:]) == len(str(b))))
```

Указания по оцениванию

Баллы

Предложена правильная программа, выдающая верный ответ для любых соответствующих условию исходных данных.
Допускается запись программы на языке, не входящем в список языков из условия. Такая программа должна использовать переменные, аналогичные заданным в условии. Если выбранный язык программирования использует типизированные переменные,

2

<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.</p> <p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора</p>	
<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"> 1) Отсутствие инициализации или неверная инициализация счётчика совпадений. 2) Выход за границы массива. 3) Изменение исходных данных в процессе решения (например, определение количества цифр с помощью деления на основание системы счисления самого элемента, а не его копии). 4) Неверная инициализация и/или построение цикла при определении количества цифр с помощью деления. Если при этом в построении обоих циклов допущена ошибка, которая всегда приводит к одинаковому изменению верного количества (например, счётчик количества цифр инициализируется единицей вместо нуля, при этом получится ответ на 1 больше правильного), то это искажение не влияет на окончательный ответ и, следовательно, не считается ошибкой. 5) Неверное определение одной или нескольких границ диапазона (с учётом использованных знаков сравнения) при решении с использованием диапазонов. 6) Неверное построение логического выражения (неверная расстановка скобок с учётом приоритетов действий в конкретном языке программирования, неверное использование операций И и ИЛИ). 7) Отсутствует вывод ответа. 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 лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может

увеличить количество камней в куче в два раза или

увеличить количество камней в куче в три раза.

Например, имея кучу из 10 камней, за один ход можно получить кучу из 20 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче превышает 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 62 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 61$.

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 следующей стратегии игрока, которые не являются для него безусловно выигрышными.

Выполните следующие задания.

Задание 1. Назовите все значения S , при которых Петя может выиграть первым ходом, причём у Пети есть **ровно один** **выигрывающий ход**.

Задание 2. Назовите все значения S , при которых Ваня может выиграть первым ходом, независимо от того, каким будет первый ход Пети. Опишите выигрышную стратегию Вани для этих значений S .

Задание 3. Назовите все значения S , при которых Петя не может выиграть первым ходом, но может выиграть вторым ходом независимо от того, как будет играть Ваня, причём в начальной позиции у Пети есть **ровно один** **выигрывающий ход**. Опишите выигрышную стратегию Пети для всех этих значений. Постройте (в виде рисунка или таблицы) дерево всех партий, возможных при этой стратегии для одного произвольного значения S . На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции. Дерево должно содержать только те партии, которые возможны при реализации выигрышной стратегии Пети.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание 1. Петя может выиграть единственным первым ходом, если $S = 21, \dots, 30$. Для выигрыша необходимо утроить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой будет 62 или более камней. При бóльших значениях S Петя тоже выигрывает первым ходом, но сделать это можно двумя способами (удвоением и утроением)

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание 2. Ваня выигрывает первым ходом при S от 11 до 20. После хода Пети в куче станет от 22 до 60 камней. В любом из этих случаев Ваня может утроить количество камней и выиграть.

Задание 3. Чтобы выиграть вторым ходом, Петя должен сделать так, чтобы после его первого хода в куче было от 11 до 20 камней. Эти позиции разобраны в задании 2, в них игрок, который будет ходить (теперь это Ваня), проигрывает. Петя может получить такую кучу единственным способом при $S = 4, 5, 7, 8, 9, 10$. При $S = 4$ и $S = 5$ Петя должен первым ходом утроить количество камней, в остальных случаях – удвоить. После этого в куче будет от 12 до 20 камней, после ответного хода Вани Петя может утроить количество камней и выиграть. Значение $S = 6$ не входит в список, так как в этом случае у Пети есть два выигрывающих хода.

В таблице изображено дерево возможных партий при описанной стратегии Пети для $S = 7$. Заключительные позиции (в них выигрывает Петя) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы). Для других значений S дерево строится аналогично.

Исходное положение	1-й ход Пети (только ход по стратегии)	1-й ход Вани (разобраны все ходы)	2-й ход Пети (только ход по стратегии)
7	$7*2 = \mathbf{14}$	$14*2 = \mathbf{28}$	$28*3 = \underline{\mathbf{84}}$
		$14*3 = \mathbf{42}$	$42*3 = \underline{\mathbf{126}}$

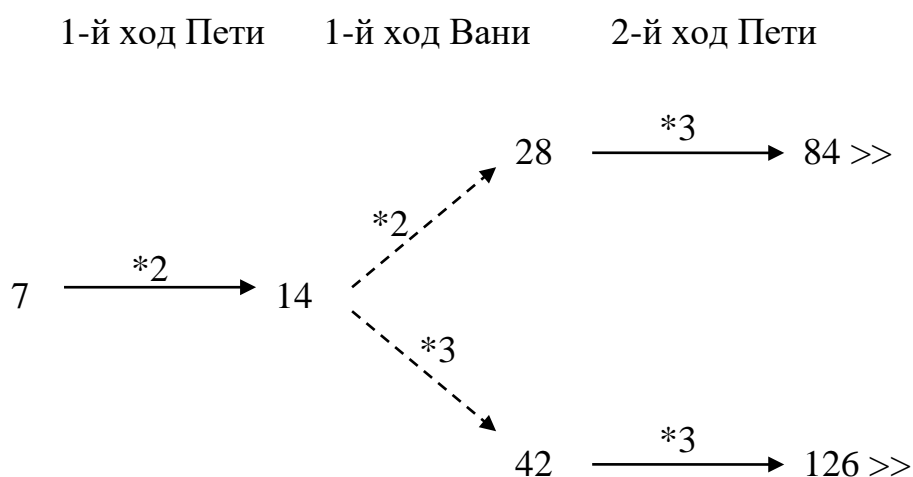


Рис. 1. Дерево всех партий, возможных при описанной стратегии Пети. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены знаком $>>$.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три 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>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом</p>	
<p>Выполнены все три задания.</p> <p>Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Выполнено задание 3. 2. Выполнены задания 1 и 2 	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Выполнено задание 1. 2. Выполнено задание 2 	1
<p>Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла</p>	0
<i>Максимальный балл</i>	3

27

Дан набор из N целых положительных чисел. Из этих чисел формируются все возможные пары (парой считаются два элемента, которые находятся на разных местах в наборе, порядок чисел в паре не учитывается), в каждой паре вычисляется сумма элементов. Необходимо определить количество пар, для которых полученная сумма делится на 9.

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

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

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 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.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$). В каждой из последующих N строк записано одно натуральное число, не превышающее 10 000.

Пример входных данных:

5
4
3
5
4
15

Пример выходных данных для приведённого выше примера входных данных:

3

Из 5 чисел можно составить 10 пар. В данном случае у трёх пар сумма делится на 9: $4 + 5$, $4 + 5$ (в наборе две четвёрки, поэтому пару $4 + 5$ можно составить двумя способами), $3 + 15$.

<p style="text-align: center;">Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Разобьём все числа исходного набора на 9 групп по значению остатка от деления на 9 и подсчитаем количество чисел в каждой группе. Сами числа можно не хранить, достаточно при вводе определить остаток от деления очередного числа на 9 и увеличить соответствующий счётчик. Таким образом, независимо от количества чисел в исходном наборе, после чтения исходных данных для хранения необходимой информации хватит массива из 9 элементов и программа получится эффективной по памяти.</p> <p>Чтобы сумма двух чисел делилась на 9, они оба должны делиться на 9 либо сумма их остатков от деления на 9 должна быть равна 9. Зная количество элементов для каждого остатка, можно определить количество пар.</p> <p>Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)</p>
<p>Пример правильной и эффективной программы на языке Паскаль</p>
<pre>var N: integer; {количество чисел} a: integer; {очередное число} d: array [0..8] of integer; {группы по остаткам} s: integer; {количество пар} i: integer; begin for i:=0 to 8 do d[i]:=0; readln(N); for i:=1 to N do begin readln(a); inc(d[a mod 9]) end; s := d[0]*(d[0]-1) div 2; for i:=1 to 4 do s := s + d[i]*d[9-i]; writeln(s) end.</pre>
<p>Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и подсчитаем количество подходящих. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растёт квадратично). Такая программа оценивается не выше 2 баллов.</p> <p>Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)</p>

Пример правильной, но неэффективной программы на языке Паскаль

```

var
  N: integer;           {количество чисел}
  a: array [1..1000] of integer; {исходные данные}
  s: integer;           {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 9 = 0
        then s := s + 1
    end
  end;
  writeln(s)
end.

```

Указания по оцениванию	Баллы
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трех синтаксических ошибок одного из следующих видов:</p> <ol style="list-style-type: none"> 1) пропущен или неверно указан знак пунктуации; 2) неверно написано или пропущено зарезервированное слово языка программирования; 3) не описана или неверно описана переменная; 4) применяется операция, недопустимая для соответствующего типа данных. <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 введенных чисел, правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в C++ или другой аналогичной структуре данных)</p> <p>Количество синтаксических ошибок («описок»), указанных в критериях на 4 балла, – не более пяти.</p>	3

Указания по оцениванию	Баллы
<p>Допускается наличие не более одной ошибки следующих видов:</p> <ol style="list-style-type: none"> 1) ошибка при инициализации или отсутствие инициализации счётчиков; 2) использование нахождения остатка вместо деления (mod вместо div в Паскале) или наоборот; 3) допущен выход за границу массива; 4) не учтены или неверно учтены пары, в которых каждый элемент делится на 9; 5) неверно составлены или учтены не все комбинации остатков; 6) неверно подсчитано количество пар для всех или некоторых комбинаций остатков (например, не выполняется деление на 2) 	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает верно, эффективно или нет. В частности, в 2 балла оцениваются переборные решения, в которых все исходные данные сохраняются в массиве, рассматриваются все возможные пары, из которых выбираются подходящие.</p> <p>Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа описывает в целом правильный алгоритм (эффективный или нет), но количество допущенных ошибок не укладывается в описанные выше ограничения</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<p><i>Максимальный балл</i></p>	4