

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 вывести минимальное натуральное K , для которого $1! + 2! + \dots + K! > A$.

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, S AS INTEGER INPUT A K = 1 F = 1 S = 0 WHILE F <= A K = K + 1 F = F * K S = S + F WEND PRINT K END </pre>	<pre> a = int(input()) k = 1 f = 1 s = 0 while f <= a: k += 1 f *= k s += f print(k) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел a, k, f, s ввод a k := 1 f := 1 s := 0 нц пока f <= a k := k + 1 f := f * k s := s + f кц вывод k кон </pre>	<pre> var a, k, f, s: integer; begin read(a); k := 1; f := 1; s := 0; while f <= a do begin k := k + 1; f := f * k; s := s + f; end; writeln(k) end. </pre>

C++

```

#include <iostream>
using namespace std;
int main(){
    int a, k, f, s;
    cin >> a;
    k = 1;
    f = 1;
    s = 0;
    while (f <= a) {
        ++k;
        f *= k;
        s += f;
    }
    cout << k;
    return 0;
}

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе $A = 8$.
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)
<ol style="list-style-type: none"> 1. При вводе $A = 8$ программа выведет число 4. 2. Минимальное $A > 10$, при котором программа выдаёт <i>неверный</i> ответ, равно 24. Программа выдаёт неверный ответ 5. При вводе чисел от 11 до 23 программа выдаёт верный ответ 4. 3. Программа содержит две ошибки. <ol style="list-style-type: none"> 1) Неверная инициализация. Значения переменных S и K не согласованы: эти переменные могут быть равны 1 или 0, но они должны быть одинаковы. В результате этой ошибки текущее значение суммы всегда оказывается на 1 меньше правильного. 2) Неверное условие цикла. Вместо суммы факториалов с заданным числом сравнивается последний вычисленный факториал

<p align="center">Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Пример исправления для языка Паскаль: Первая ошибка, способ 1: <code>k := 1;</code> Исправленная строка: <code>k := 0;</code> Первая ошибка, способ 2: <code>s := 0;</code> Исправленная строка: <code>s := 1;</code> Вторая ошибка: <code>while f <= a do begin</code> Исправленная строка: <code>while s <= a do begin</code> 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</p>

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 другой ошибки получается правильная программа 	

Указания по оцениванию	Баллы
Выполнены все три необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: <ul style="list-style-type: none"> а. выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной; б. выполнены два первых действия, найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной; в. выполнено одно из первых двух действий, найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной 	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев: <ul style="list-style-type: none"> а. выполнены два первых действия. При этом несущественно, насколько правильно выполнено третье действие; б. 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; в. 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 найти в этом массиве количество элементов, которые кратны 3, а их десятичная запись заканчивается цифрой 7, и заменить каждый из таких элементов на это количество.

Напишите на одном из языков программирования программу для решения этой задачи. В качестве результата программа должна вывести изменённый массив, по одному элементу в строке. Например, для исходного массива из 5 элементов 12 17 27 117 48 программа должна вывести числа 12 17 2 2 48, по одному числу в строке.

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 I, K, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END </pre>	<pre> # кроме уже указанных # допускается использование # целочисленных переменных # k, 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 i, k, 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; i, k, 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 i, k, 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.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)	
Задача решается в два прохода: на первом проходе подсчитывается количество удовлетворяющих условиям элементов, на втором каждый такой элемент заменяется на найденное количество и выводится. Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод	
Пример правильной программы на языке Паскаль	
<pre> k := 0; for i:=1 to N do begin if (a[i] mod 3 = 0) and (a[i] mod 10 = 7) then k := k+1; end; for i:=1 to N do begin if (a[i] mod 3 = 0) and (a[i] mod 10 = 7) then a[i] := k; writeln(a[i]) end;</pre>	
При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка	
Пример правильной программы на языке Python	
<pre> k = sum (1 for m in a if m%3 == 0 and m%10 == 7) for i in range(0,n): if a[i]%3 == 0 and a[i]%10 == 7: a[i] = k print(a[i])</pre>	
Использовать описанную выше возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на языке Паскаль	

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 только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора	2

Указания по оцениванию	Баллы
<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; 2) выход за границы массива; 3) неверно определяется кратность числа 3; 4) на кратность 3 проверяется не элемент, а его индекс; 5) неверно определяется последняя цифра десятичной записи числа; 6) неверное изменение счётчика при нахождении очередного соответствующего условию элемента; 7) исходный массив не изменяется; 8) неверное построение логического выражения (неверная расстановка скобок с учётом приоритетов действий в конкретном языке программирования, неверное использование операций И и ИЛИ); 9) полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы); 10) используется переменная, не объявленная в разделе описания переменных; 11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно 	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

добавить в кучу один камень или

добавить в кучу два камня или

увеличить количество камней в куче в два раза.

Например, имея кучу из 10 камней, за один ход можно получить кучу из 11, 12 или 20 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче превышает 37. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 38 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 37$.

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.

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

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

Задание 2.

Укажите три значения S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.

Задание 3.

Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Дерево не должно содержать партий, невозможных при реализации выигрышающим 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 единственным способом (увеличив количество камней в 2 раза), если $S = 19, \dots, 35$. При меньших значениях S за один ход нельзя получить кучу, в которой будет 38 или более камней. При $S = 36$ и $S = 37$ у Пети есть более одного выигрывающего хода.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 18$ камней. Тогда после первого хода Пети в куче будет 19, 20 или 36 камней. Во всех случаях Ваня увеличивает количество камней в 2 раза и выигрывает в один ход.

Задание 2.

Возможные значения S : 9, 16, 17. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 18 камней (при $S = 9$ он удваивает количество камней; при $S = 16$ добавляет 2 камня; при $S = 17$ добавляет 1 камень). Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (в данном случае это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

Задание 3.

Возможное значение S : 15. После первого хода Пети в куче будет 16, 17 или 30 камней. Если в куче станет 30 камней, то Ваня увеличит количество камней в 2 раза и выиграет своим первым ходом. Ситуация, когда в куче 16 или 17 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исходное положение	Положения после очередных ходов			
	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)
15	$15 + 1 = \mathbf{16}$	$16 + 2 = \mathbf{18}$	$18 + 1 = \mathbf{19}$	$19 * 2 = \mathbf{38}>>$
			$18 + 2 = \mathbf{20}$	$20 * 2 = \mathbf{40}>>$
			$18 * 2 = \mathbf{36}$	$36 * 2 = \mathbf{72}>>$
	$15 + 2 = \mathbf{17}$	$17 + 1 = \mathbf{18}$	$18 + 1 = \mathbf{19}$	$19 * 2 = \mathbf{38}>>$
			$18 + 2 = \mathbf{20}$	$20 * 2 = \mathbf{40}>>$
			$18 * 2 = \mathbf{36}$	$36 * 2 = \mathbf{72}>>$
	$15 * 2 = \mathbf{30}$	$30 * 2 = \mathbf{60}>>$		

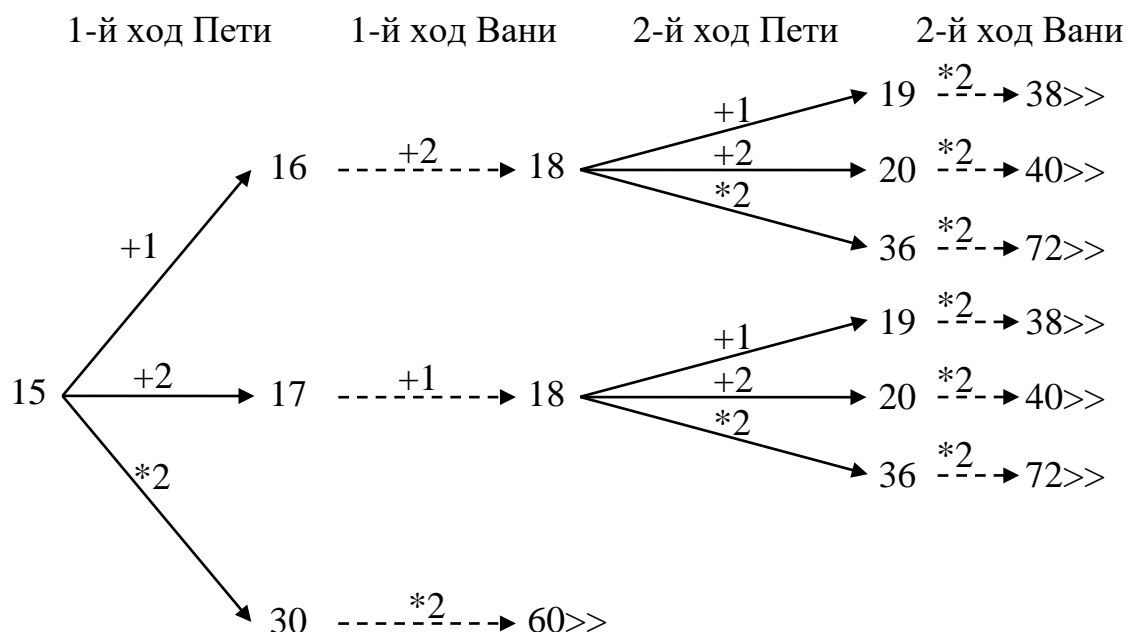


Рис. 1. Дерево всех партий, возможных при описанной стратегии Пети. Ходы Пети показаны сплошными стрелками, ходы Вани показаны пунктирными стрелками. Заключительные позиции обозначены знаком >>.

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, вершины дерева, соответствующие одной и той же позиции, на рисунке могут быть «склеены». Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

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	2

Указания по оцениванию	Баллы
1) Выполнено задание 3. 2) Выполнены задания 1 и 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
Максимальный балл	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 8.

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

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

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

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

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

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

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

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

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
5
7
11
1

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 8: $1 + 7$, $1 + 7$ (в наборе две единицы, поэтому пару $1 + 7$ можно составить двумя способами), $5 + 11$.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Чтобы сумма двух чисел делилась на 8, они оба должны делиться на 8 либо сумма их остатков от деления на 8 должна быть равна 8.

Разобьём все числа исходного набора на 8 групп по значению остатка от деления на 8 и подсчитаем количество чисел в каждой группе. Сами числа можно не хранить, достаточно при вводе определить остаток от деления очередного числа на 8 и увеличить соответствующий счётчик. Таким образом, независимо от количества чисел в исходном наборе после чтения исходных данных для хранения необходимой информации хватит массива из 8 элементов, и программа получится эффективной по памяти.

Зная количество элементов для каждого остатка, можно определить количество подходящих пар. Для чисел с остатками 0 (числа, кратные 8) и 4 оба числа в паре должны быть в одной и той же группе. Если в группе k элементов, то количество пар равно $k(k-1)/2$. Для чисел с другими остатками группы разбиваются на пары (1 и 7, 2 и 6, 3 и 5), количество пар равно произведению количества элементов в соответствующих группах.

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

Пример правильной и эффективной программы на языке Паскаль

```
var
    N: integer;      {количество чисел}
    a: integer;      {очередное число}
    d: array [0..7] of integer; {группы по остаткам}
    s: integer;      {количество пар}
    i: integer;

begin
    for i:=0 to 7 do d[i]:=0;
    readln(N);
    for i:=1 to N do begin
        readln(a);
        inc(d[a mod 8])
    end;
    s := (d[0]*(d[0]-1) + d[4]*(d[4]-1)) div 2;
    for i:=1 to 3 do s := s + d[i]*d[8-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 двух 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 8 = 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 введённых чисел, правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p> <p>Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются 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 8; 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 для остатка 4) 	3

Указания по оцениванию	Баллы
<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
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 вывести минимальное натуральное число K , для которого $1! + 2! + \dots + K! > A$.

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, S AS INTEGER INPUT A K = 1 F = 1 S = 1 WHILE S <= A F = F * K K = K + 1 S = S + F WEND PRINT K END </pre>	<pre> a = int(input()) k = 1 f = 1 s = 1 while s <= a: f *= k k += 1 s += f print(k) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел a, k, f, s ввод a k := 1 f := 1 s := 1 нц пока s <= a f := f * k k := k + 1 s := s + f кц вывод k кон </pre>	<pre> var a, k, f, s: integer; begin read(a); k := 1; f := 1; s := 1; while s <= a do begin f := f * k; k := k + 1; s := s + f; end; writeln(k); end. </pre>

C++

```
#include <iostream>
using namespace std;
int main(){
    int a, k, f, s;
    cin >> a;
    k = 1;
    f = 1;
    s = 1;
    while (s <= a) {
        f *= k;
        ++k;
        s += f;
    }
    cout << k;
    return 0;
}
```

Последовательно выполните следующее.

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

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

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

Содержание верного ответа
(допускаются иные формулировки ответа, не искажающие его смысла)
<ol style="list-style-type: none"> 1. При вводе $A = 6$ программа выведет число 4. 2. Минимальное $A > 10$, при котором программа выдаёт <i>верный</i> ответ, равно 33. Программа выдаёт верный ответ 5. При вводе чисел от 11 до 32 программа выдаёт ответ 5 вместо верного ответа 4. 3. Программа содержит две ошибки. <ol style="list-style-type: none"> 1) Неверная инициализация. Значения переменных K и S не согласованы: эти переменные могут быть равны 1 и 0 или 2 и 1, но не 1 и 1. В результате этой ошибки текущее значение суммы всегда оказывается на 1 больше правильного

<p align="center">Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>2) Неверный порядок действий в цикле. В результате факториал и сумма факториалов вычисляются для предыдущего значения <i>K</i>. Поскольку менять строки местами не разрешается, для исправления ошибки нужно внести коррективы при выводе ответа.</p> <p>Пример исправления для языка Паскаль:</p> <p>Первая ошибка, способ 1: <code>k := 1;</code> Исправленная строка: <code>k := 2;</code></p> <p>Первая ошибка, способ 2: <code>s := 1;</code> Исправленная строка: <code>s := 0;</code></p> <p>Вторая ошибка: <code>writeln(k)</code> Исправленная строка: <code>writeln(k-1)</code></p> <p>В программах на других языках ошибочные строки и их исправления аналогичны.</p> <p>Незначительной опiskой, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления</p>

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить три действия.</p> <p>1) Указать 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)</p>	

Указания по оцениванию	Баллы
<p>Ошибка считается исправленной, если выполнены оба следующих условия:</p> <ul style="list-style-type: none"> а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа 	
Выполнены все три необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <ul style="list-style-type: none"> а. выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной; б. выполнены два первых действия, найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной; в. 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> <ul style="list-style-type: none"> а. выполнены два первых действия. При этом несущественно, насколько правильно выполнено третье действие; б. 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; в. 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 найти в этом массиве количество элементов, которые кратны 7, а их десятичная запись заканчивается цифрой 3, и заменить каждый из таких элементов на это количество.

Напишите на одном из языков программирования программу для решения этой задачи. В качестве результата программа должна вывести изменённый массив, по одному элементу в строке. Например, для исходного массива из 5 элементов 14 13 63 203 49 программа должна вывести числа 14 13 2 2 49, по одному числу в строке.

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 I, K, M AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre># кроме уже указанных # допускается использование # целочисленных переменных # k, 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 i, k, 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; i, k, 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 i, k, 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.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)	
Задача решается в два прохода: на первом проходе подсчитывается количество удовлетворяющих условиям элементов, на втором каждый такой элемент заменяется на найденное количество и выводится. Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод	
Пример правильной программы на языке Паскаль	
<pre> k := 0; for i:=1 to N do begin if (a[i] mod 7 = 0) and (a[i] mod 10 = 3) then k := k+1; end; for i:=1 to N do begin if (a[i] mod 7 = 0) and (a[i] mod 10 = 3) then a[i] := k; writeln(a[i]) end; </pre>	
При использовании языка Python первый проход можно записать в одну строку, используя специальные средства этого языка.	
Пример правильной программы на языке Python	
<pre> k = sum (1 for m in a if m%7 == 0 and m%10 == 3) for i in range(0,n): if a[i]%7 == 0 and a[i]%10 == 3: a[i] = k print(a[i]) </pre>	
Использовать описанную выше возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на языке Паскаль	

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

Указания по оцениванию	Баллы
<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; 2) выход за границы массива; 3) неверно определяется кратность числа 7; 4) на кратность 7 проверяется не элемент, а его индекс; 5) неверно определяется последняя цифра десятичной записи числа; 6) неверное изменение счётчика при нахождении очередного соответствующего условию элемента; 7) исходный массив не изменяется; 8) неверное построение логического выражения (неверная расстановка скобок с учётом приоритетов действий в конкретном языке программирования, неверное использование операций И и ИЛИ); 9) полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы); 10) используется переменная, не объявленная в разделе описания переменных; 11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно 	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

добавить в кучу один камень или

добавить в кучу два камня или

увеличить количество камней в куче в два раза.

Например, имея кучу из 10 камней, за один ход можно получить кучу из 11, 12 или 20 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче превышает 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 42 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 41$.

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 следующего игрока, которые не являются для него безусловно выигрышными.

Выполните следующие задания.

Задание 1.

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

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

Задание 2.

Укажите три значения S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.

Задание 3.

Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Дерево не должно содержать партий, невозможных при реализации выигрышающим 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 единственным способом (увеличив количество камней в 2 раза), если $S = 21, \dots, 39$. При меньших значениях S за один ход нельзя получить кучу, в которой будет 42 или более камней. При $S = 40$ и $S = 41$ у Пети есть более одного выигрывающего хода.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 20$ камней. Тогда после первого хода Пети в куче будет 21, 22 или 40 камней. Во всех случаях Ваня увеличивает количество камней в 2 раза и выигрывает в один ход.

Задание 2.

Возможные значения S : 10, 18, 19. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 20 камней (при $S = 10$ он удваивает количество камней; при $S = 18$ добавляет 2 камня; при $S = 19$ добавляет 1 камень). Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (в данном случае это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выиграет.

Задание 3.

Возможное значение S : 17. После первого хода Пети в куче будет 18, 19 или 34 камня. Если в куче станет 34 камня, то Ваня увеличит количество камней в 2 раза и выиграет своим первым ходом. Ситуация, когда в куче 18 или 19 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчеркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исходное положение	Положения после очередных ходов			
	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)
17	$17 + 1 = \mathbf{18}$	$18 + 2 = \mathbf{20}$	$20 + 1 = \mathbf{21}$	$21 * 2 = \mathbf{42}>>$
			$20 + 2 = \mathbf{22}$	$22 * 2 = \mathbf{44}>>$
			$20 * 2 = \mathbf{40}$	$40 * 2 = \mathbf{80}>>$
	$17 + 2 = \mathbf{19}$	$19 + 1 = \mathbf{20}$	$20 + 1 = \mathbf{21}$	$21 * 2 = \mathbf{42}>>$
			$20 + 2 = \mathbf{22}$	$22 * 2 = \mathbf{44}>>$
			$20 * 2 = \mathbf{40}$	$40 * 2 = \mathbf{80}>>$
	$17 * 2 = \mathbf{34}$	$34 * 2 = \mathbf{68}>>$		

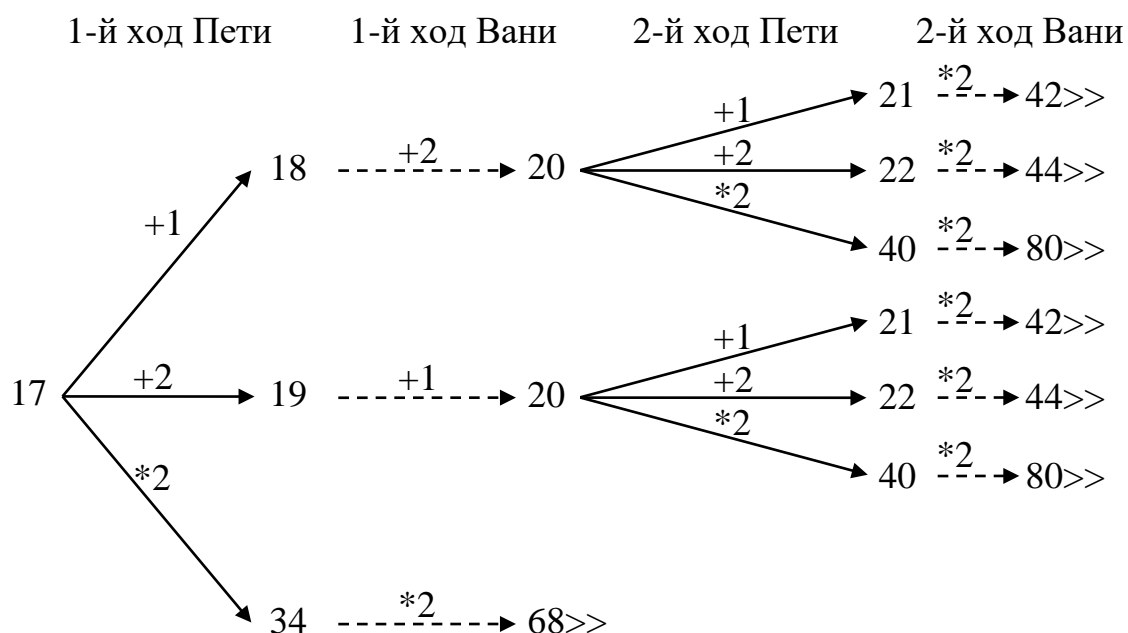


Рис. 1. Дерево всех партий, возможных при описанной стратегии Пети. Ходы Пети показаны сплошными стрелками, ходы Вани показаны пунктирными стрелками. Заключительные позиции обозначены знаком >>.

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, вершины дерева, соответствующие одной и той же позиции, на рисунке могут быть «склеены». Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

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

Указания по оцениванию	Баллы
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. а. Выполнено задание 3. б. Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий. а. Выполнено задание 1. б. Выполнено задание 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
Максимальный балл	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 10.

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

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

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

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

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

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

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

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

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
5
6
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 10: $4 + 6$, $4 + 6$ (в наборе две четвёрки, поэтому пару $4 + 6$ можно составить двумя способами), $5 + 15$.

<p style="text-align: center;">Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Чтобы сумма двух чисел делилась на 10, они оба должны делиться на 10 либо сумма их остатков от деления на 10 должна быть равна 10.</p> <p>Разобьём все числа исходного набора на 10 групп по значению остатка от деления на 10 и подсчитаем количество чисел в каждой группе. Сами числа можно не хранить, достаточно при вводе определить остаток от деления очередного числа на 10 и увеличить соответствующий счётчик. Таким образом, независимо от количества чисел в исходном наборе после чтения входных данных для хранения необходимой информации хватит массива из 10 элементов, и программа получится эффективной по памяти.</p> <p>Зная количество элементов для каждого остатка, можно определить количество пар. Для чисел с остатками 0 (числа, кратные 10) и 5 оба числа в паре должны быть в одной и той же группе. Если в группе k элементов, то количество пар равно $k(k-1)/2$. Для чисел с другими остатками группы разбиваются на пары (1 и 9, 2 и 8, 3 и 7, 4 и 6), количество пар равно произведению количества элементов в соответствующих группах.</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..9] of integer; {группы по остаткам}
  s: integer;      {количество пар}
  i: integer;

begin
  for i:=0 to 9 do d[i]:=0;
  readln(N);
  for i:=1 to N do begin
    readln(a);
    inc(d[a mod 10])
  end;
  s := (d[0]*(d[0]-1) + d[5]*(d[5]-1)) div 2;
  for i:=1 to 4 do s := s + d[i]*d[10-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 двух 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 10 = 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 10; 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 для остатка 5) 	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>	4