

Критерии оценивания заданий с развёрнутым ответом**24**

Дано натуральное число N ($N \leq 10^9$). Необходимо найти и вывести наибольшую нечётную цифру в десятичной записи N или вывести сообщение «NO», если таких цифр нет.

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

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

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main(){ int N, d, m, t; cin >> N; t = 0; m = t; while (N > 1) { d = N % 10; if (d%2 == 1 d>m) m = d; N = N /10; } if (m == t) cout >> "NO"; else cout << m; return 0; }</pre>	<pre>алг нач цел N, d, m, t ввод N t := 0 m := t нц пока N > 1 d := mod(N,10) если mod(d,2)=1 или d>m то m := d все N := div(N,10) кц если m = t то вывод "NO" иначе вывод m все кон</pre>
Python	Паскаль
<pre>N = int(input()) t = 0 m = t while N > 1: d = N % 10 if d%2 == 1 or d > m: m = d N = N // 10 if m == t: print("NO") else: print(m)</pre>	<pre>var N, d, m, t: integer; begin read(N); t := 0; m := t; while N > 1 do begin d := N mod 10; if (d mod 2 = 1) or (d>m) then m := d; N := N div 10 end; if m = t then writeln('NO') else writeln(m) end.</pre>

Бейсик

```

DIM N, D, M, T AS INTEGER
INPUT N
T = 0
M = T
WHILE N > 1
    D = N MOD 10
    IF D MOD 2 = 1 OR D > M THEN
        M = D
    END IF
WEND
IF M = T THEN
    PRINT "NO"
ELSE
    PRINT M
END IF
END

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе $N=143$.
2. Какое **наименьшее** число может быть выведено при работе этой программы? Приведите пример числа N , при вводе которого программа выведет такой 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)

1. При вводе $N=143$ программа выведет число 4.
2. Наименьшее число, которое может вывести программа, равно 1.
Возможные значения N : 11, 101, 1010.
3. Программа содержит две ошибки.
 - 1) Неверное условие цикла. В результате, если первая цифра числа равна 1, она не обрабатывается, хотя может оказаться верным ответом.
 - 2) Неверное условие проверки цифры, вместо логического И использовано ИЛИ. В результате при последовательной проверке цифр как текущий максимум фиксируется любая (не обязательно наибольшая) нечётная цифра либо чётная, которая больше текущего максимума.

Пример исправления для языка Паскаль:**Первая ошибка:**

```
while N > 1 do begin
```

Исправленная строка, способ 1:

```
while N > 0 do begin
```

Исправленная строка, способ 2:

```
while N >= 1 do begin
```

Вторая ошибка:

```
if (d mod 2 = 1) or (d>m) then
```

Исправленная строка:

```
if (d mod 2 = 1) and (d>m) then
```

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. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать минимально возможное значение ответа программы и исходные данные, при которых выдаётся такой ответ. Это действие считается выполненным, если указано верное значение минимума и значение исходного числа, при котором выводится такой минимум. Экзаменуемый не обязан пояснять, как получены эти числа, и доказывать, что они удовлетворяют 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 	

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. <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
Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев: <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>	3

25

Дан массив, содержащий 2018 положительных целых чисел, не превышающих 30 000. Необходимо уменьшить все чётные элементы этого массива на одно и то же значение, при этом минимальный из них должен стать равным 2.

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 112 87 27 95 148 программа должна вывести числа 2 87 27 95 38 по одному числу в строке (все чётные элементы уменьшены, минимальный из них стал равен 2).

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 определяется минимальный чётный элемент, на втором производится корректировка и вывод элементов. Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод	
Пример правильной программы на языке Паскаль	
<pre> m := 30001; for i:=1 to N do begin if (a[i] mod 2 = 0) and (a[i] < m) then m := a[i]; end; for i:=1 to N do begin if a[i] mod 2 = 0 then a[i] := a[i] + 2 - m; writeln(a[i]) end;</pre>	
При использовании языка Python первый проход можно записать в одну строку, применяя специальные средства этого языка. В приведённом ниже примере использован параметр default, действующий в версии Python 3.4 и более поздних. При использовании ранних версий Python необходимо отдельно проверять наличие в массиве удовлетворяющих условию элементов	
Пример правильной программы на языке Python	
<pre> m = min((k for k in a if k%2 == 0), default=0) for i in range(0,n): if a[i] % 2 == 0: a[i] = a[i] + 2 - m 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.	2

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	
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку).</p> <p>1) Отсутствие инициализации или неверная инициализация минимума.</p> <p>2) Неверное определение минимума (например, поиск максимума вместо минимума).</p> <p>3) Определение минимума среди всех элементов, без учета указанного в условии ограничения.</p> <p>4) Выход за границы массива.</p> <p>5) Исходный массив не изменяется.</p> <p>6) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться.</p> <p>7) Неверное изменение (минимальный из изменённых элементов не равен 2).</p> <p>8) Неверное построение логического выражения (неверная расстановка скобок с учётом приоритетов действий в конкретном языке программирования, неверное использование операций И и ИЛИ).</p> <p>9) Полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы).</p> <p>10) Используется переменная, не объявленная в разделе описания переменных.</p> <p>11) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно</p>	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

увеличить количество камней в куче в два раза.

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

Игра завершается в тот момент, когда количество камней в куче превышает 53. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 54 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 53$.

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 два 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 два раза), если $S = 27, \dots, 52$. При меньших значениях S за один ход нельзя получить кучу, в которой будет 54 или более камней. При $S = 53$ у Пети есть более одного выигрывающего хода.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 26$ камней. Тогда после первого хода Пети в куче будет 27 или 52 камня. В обоих случаях Ваня увеличивает количество камней в 2 раза и выигрывает в один ход.

Задание 2.

Возможные значения S : 13, 25. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 26 камней (при $S = 13$ он удваивает количество камней, при $S = 25$ 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 : 24. После первого хода Пети в куче будет 25 или 48 камней. Если в куче станет 48 камней, то Ваня увеличит количество камней в 2 раза и выиграет своим первым ходом. Ситуация, когда в куче 25 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)
24	$24 + 1 = \mathbf{25}$	$25 + 1 = \mathbf{26}$	$26 + 1 = \mathbf{27}$	$27 * 2 = \mathbf{54}>>$
			$26 * 2 = \mathbf{52}$	$52 * 2 = \mathbf{104}>>$
	$24 * 2 = \mathbf{48}$	$48 * 2 = \mathbf{96}>>$		

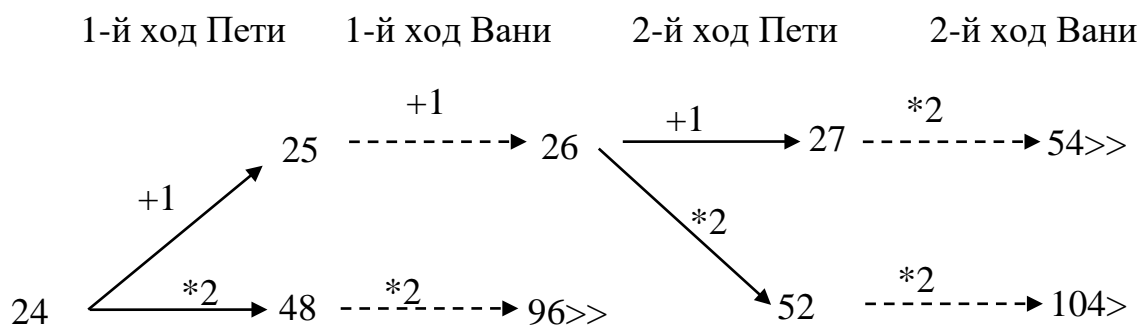


Рис. 1. Дерево всех партий, возможных при описанной стратегии Пети. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены знаком >>.

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. 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. <ul style="list-style-type: none"> – Выполнено задание 3. – Выполнены задания 1 и 2 	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий. <ul style="list-style-type: none"> – Выполнено задание 1. – Выполнено задание 2 	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

27

Дан набор из N целых положительных чисел. Из них нужно выбрать и вывести два числа так, чтобы их сумма была нечётна, а произведение делилось на 3 и при этом было максимально возможным. Выбранные числа можно выводить в любом порядке. Если есть несколько подходящих пар, можно выбрать любую из них. Если подходящих пар нет, нужно вывести 0. 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 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 100.

Пример входных данных:

5
1
2
3
4
5

Пример выходных данных для приведённого выше примера входных данных:

3 4

Пояснение. Из 5 чисел можно составить 10 пар. В данном случае условиям удовлетворяют две пары: (2, 3) и (3, 4). Суммы чисел в этих парах (5 и 7) нечётны, а произведения (6 и 12) делятся на 3. У всех остальных пар как минимум одно из этих условий не выполняется. Из двух возможных пар выводим ту, в которой больше произведение элементов.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Разобьём все числа исходного набора на 4 группы в зависимости от их чётности и делимости на 3:

группа m_1 : нечётные числа, не кратные 3;

группа m_2 : чётные числа, кратные 3;

группа m_3 : нечётные числа, кратные 3;

группа m_6 : чётные числа, кратные 3.

Чтобы сумма двух чисел была нечётной, одно из них должно быть чётным, а другое – нечётным. Чтобы произведение двух чисел делилось на 3, хотя бы одно из этих чисел должно делиться на 3. Таким образом, нужно выбрать два числа из групп m_1 и m_6 , или из групп m_2 и m_3 , или из групп m_3 и m_6 . Чтобы получить пару с максимальным произведением, достаточно сохранить максимальный элемент из каждой группы, сравнить соответствующие произведения и выбрать из них наибольшее.

Сами числа при этом можно не хранить, достаточно при вводе определить остаток от деления очередного числа на 2 и на 3, сравнить число с текущим максимумом соответствующей группы и при необходимости обновить этот максимум и увеличить соответствующий счётчик. Таким образом, независимо от количества чисел в исходном наборе, после чтения исходных данных для хранения необходимой информации хватит 4 переменных, и программа получится эффективной по памяти.

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}
  m1: integer;     {нечётные, не кратные 3}
  m2: integer;     {чётные, не кратные 3}
  m3: integer;     {нечётные, кратные 3}
  m6: integer;     {чётные, кратные 3}
  x, y: integer;   {выбранная пара}
  i: integer;

begin
  m1:=0; m2:=0; m3:=0; m6:=0;
  readln(N);
  for i:=1 to N do begin
    readln(a);
    if a mod 2 = 0 then begin
      if a mod 3 = 0 then begin
        if a>m6 then m6 := a
      end
    else begin
      if a>m2 then m2 := a
    end
  end
  else begin
    if a mod 3 = 0 then begin
      if a>m3 then m3 := a
    else begin
      if a>m1 then m1 := a
    end
  end
  end;
  x:=m1; y:=m6;
  if m2*m3 > x*y then begin
    x:=m2; y:=m3;
  end
  if m3*m6 > x*y then begin
    x:=m3; y:=m6;
  end
  if x*y = 0
  then writeln(0)
  else writeln(x, ' ', y);
end.
```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары, выберем из них требуемую. 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}
  x, y: integer;   {выбранная пара}
  i, j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  x :=0; y:=0;
  for i := 1 to N-1 do begin
    for j := i+1 to N do begin
      if ((a[i]+a[j]) mod 2 = 1) and
        ((a[i]*a[j]) mod 3 = 0) and
        (a[i]*a[j] > x*y)
      then begin
        x:=a[i]; y:=a[j]
      end
    end
  end
  end;
  if x = 0
  then writeln(0)
  else writeln(x, ' ', y);
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 групповых максимумов; 2) использование деления вместо нахождения остатка (div вместо mod в Паскале) или наоборот; 3) ошибка при построении системы проверок, в результате которой группы некоторых чисел определяются неверно; 4) учитываются не все возможные комбинации, дающие требуемый результат (например, не учитываются пары, в которых оба элемента кратны 3); 5) неверно определяется пара с максимальным произведением; 6) неверная обработка ситуации, в которой в наборе нет требуемых пар (например, вывод в этой ситуации двух чисел, как минимум одно из которых равно нулю); 7) неверный вывод ответа (например, вывод произведения вместо сомножителей) 	3

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. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла	2
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

Критерии оценивания заданий с развёрнутым ответом**24**

Дано натуральное число N ($N \leq 10^9$). Необходимо найти и вывести наибольшую чётную цифру в десятичной записи N или вывести сообщение «NO», если таких цифр нет.

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

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

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main(){ int N, d, m, t; cin >> N; t = 0; m = t; while (N > 1) { d = N % 10; if (d%2 == 0 && d>m) m = m + d; N = N / 10; } if (m == t) cout >> "NO"; else cout << m; return 0; }</pre>	<pre>алг нач цел N, d, m, t ввод N t := 0 m := t нц пока N > 1 d := mod(N,10) если mod(d,2)=0 и d>m то m := m + d все N := div(N,10) кц если m = t то вывод "NO" иначе вывод m все кон</pre>
Python	Паскаль
<pre>N = int(input()) t = 0 m = t while N > 1: d = N % 10 if d%2 == 0 and d > m: m = m + d N = N // 10 if m == t: print("NO") else: print(m)</pre>	<pre>var N, d, m, t: integer; begin read(N); t := 0; m := t; while N > 1 do begin d := N mod 10; if (d mod 2 = 0) and (d>m) then m := m + d; N := N div 10 end; if m = t then writeln('NO') else writeln(m) end.</pre>

Бейсик

```

DIM N, D, M, T AS INTEGER
INPUT N
T = 0
M = T
WHILE N > 1
    D = N MOD 10
    IF D MOD 2 = 0 AND D > M THEN
        M = M + D
    END IF
WEND
IF M = T THEN
    PRINT "NO"
ELSE
    PRINT M
END IF
END

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе $N=864$.
2. Какое **наибольшее** число может стать результатом работы этой программы? Приведите пример числа N , при вводе которого программа выведет такой 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)

1. При вводе $N=864$ программа выведет число 10.
2. Наибольший возможный результат работы программы равен 14. Такой результат будет выведен, например, при $N = 86$ и $N = 842$.
3. Программа содержит две ошибки.
 - 1) Неверное начальное значение максимума. В результате не различаются ситуации, когда в числе вообще нет чётных цифр и когда все чётные цифры – нули.
 - 2) Вместо замены максимального значения происходит вычисление суммы.

Пример исправления для языка Паскаль:***Первая ошибка:***

```
t := 0;
```

Исправленная строка:

```
t := -1;
```

Вместо -1 возможно использование любого другого отрицательного числа.

Вторая ошибка:

```
m := m + d;
```

Исправленная строка:

```
m := d;
```

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.

Условие цикла в строке

```
while N > 1 do begin
```

похоже на ошибку, но ошибкой не является, так как не приводит к неверным результатам. Это условие приводит к тому, что если первая цифра числа равна 1, эта цифра не будет рассматриваться, но так как 1 – нечётная цифра, на результат это не влияет.

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. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.</p> <p>2. Указать максимально возможное значение ответа программы и исходные данные, при которых выдаётся такой ответ. Это действие считается выполненным, если указано верное значение максимума и значение исходного числа, при котором выводится такой максимум. Экзаменуемый не обязан пояснять, как получены эти числа, и доказывать, что они удовлетворяют 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 	
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"> 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

Не выполнены условия, позволяющие поставить 2 или 3 балла. При этом имеет место один из следующих случаев. <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 уменьшить все нечётные элементы этого массива на одно и то же значение, при этом минимальный из них должен стать равным 1.

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 112 87 27 95 148 программа должна вывести числа 112 61 1 69 148 по одному числу в строке (все нечётные элементы уменьшены, минимальный из них стал равен 1).

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)
<p>Задача решается в два прохода: на первом проходе определяется минимальный нечётный элемент, на втором производится корректировка и вывод элементов.</p> <p>Возможно решение в три прохода, когда на втором проходе выполняется только замена значений, а на третьем – вывод</p>

Пример правильной программы на языке Паскаль

```

m := 30001;
for i:=1 to N do begin
    if (a[i] mod 2 = 1) and (a[i] < m)
        then m := a[i];
end;
for i:=1 to N do begin
    if a[i] mod 2 = 1
        then a[i] := a[i] + 1 - m;
    writeln(a[i])
end;

```

При использовании языка Python первый проход можно записать в одну строку, применяя специальные средства этого языка. В приведённом ниже примере использован параметр `default`, действующий в версии Python 3.4 и более поздних. При использовании ранних версий Python необходимо отдельно проверять наличие в массиве удовлетворяющих условию элементов

Пример правильной программы на языке Python

```

m = min((k for k in a if k%2 == 1), default=0)
for i in range(0,n):
    if a[i] % 2 == 1:
        a[i] = a[i] + 1 - m
    print(a[i])

```

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
<p>Предложена правильная программа, выдающая верный ответ для любых соответствующих условию исходных данных.</p> <p>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.</p> <p>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>	2

<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку).</p> <ol style="list-style-type: none"> 1) Отсутствие инициализации или неверная инициализация минимума. 2) Неверное определение минимума (например, поиск максимума вместо минимума). 3) Определение минимума среди всех элементов, без учёта указанного в условии ограничения. 4) Выход за границы массива. 5) Исходный массив не изменяется. 6) Изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться. 7) Неверное изменение (минимальный из изменённых элементов не равен 1). 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

увеличить количество камней в куче в два раза.

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

Игра завершается в тот момент, когда количество камней в куче превышает 45. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 46 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 45$.

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 два 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 два раза), если $S = 23, \dots, 43$. При меньших значениях S за один ход нельзя получить кучу, в которой будет 46 или более камней. При $S = 44$ у Пети есть более одного выигрывающего хода.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 22$ камня. Тогда после первого хода Пети в куче будет 23 или 44 камня. В обоих случаях Ваня увеличивает количество камней в 2 раза и выигрывает в один ход.

Задание 2.

Возможные значения S : 11, 21. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 22 камней (при $S = 11$ он удваивает количество камней, при $S = 21$ 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 : 20. После первого хода Пети в куче будет 21 или 40 камней. Если в куче станет 40 камней, то Ваня увеличит количество камней в 2 раза и выиграет своим первым ходом. Ситуация, когда в куче 21 камень, 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)
20	$20 + 1 = \mathbf{21}$	$21 + 1 = \mathbf{22}$	$22 + 1 = \mathbf{23}$	$23 * 2 = \mathbf{46}>>$
			$22 * 2 = \mathbf{44}$	$44 * 2 = \mathbf{88}>>$
	$20 * 2 = \mathbf{40}$	$40 * 2 = \mathbf{80}>>$		

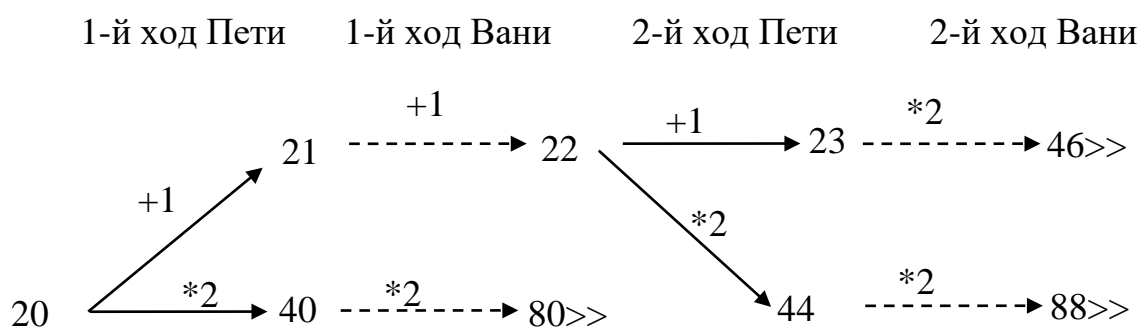


Рис. 1. Дерево всех партий, возможных при описанной стратегии Пети. Ходы Пети показаны сплошными стрелками, ходы Вани – пунктирными стрелками. Заключительные позиции обозначены знаком >>.

Примечание для эксперта. Дерево всех партий может быть изображено в виде таблицы или ориентированного графа – так, как показано на рисунке, или другим способом. 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. <ul style="list-style-type: none"> – Выполнено задание 3. – Выполнены задания 1 и 2 	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий. <ul style="list-style-type: none"> – Выполнено задание 1. – 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 них нужно выбрать и вывести два числа так, чтобы их сумма была нечётна, а произведение делилось на 5 и при этом было максимально возможным. Выбранные числа можно выводить в любом порядке. Если есть несколько подходящих пар, можно выбрать любую из них. Если подходящих пар нет, нужно вывести 0.

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 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 100.

Пример входных данных:

5
1
2
4
5
7

Пример выходных данных для приведённого выше примера входных данных:

4 5

Пояснение. Из 5 чисел можно составить 10 пар. В данном случае условиям удовлетворяют две пары: (2, 5) и (4, 5). Суммы чисел в этих парах (7 и 11) нечётны, а произведения (10 и 20) делятся на 5. У всех остальных пар как минимум одно из этих условий не выполняется. Из двух возможных пар выводим ту, в которой больше произведение элементов.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Разобьём все числа исходного набора на 4 группы в зависимости от их чётности и делимости на 3:

группа m_1 : нечётные числа, не кратные 5;

группа m_2 : чётные числа, кратные 5;

группа m_5 : нечётные числа, кратные 5;

группа m_{10} : чётные числа, кратные 5.

Чтобы сумма двух чисел была нечётной, одно из них должно быть чётным, а другое – нечётным. Чтобы произведение двух чисел делилось на 5, хотя бы одно из этих чисел должно делиться на 5. Таким образом, нужно выбрать два числа из групп m_1 и m_{10} , или из групп m_2 и m_5 , или из групп m_5 и m_{10} . Чтобы получить пару с максимальным произведением, достаточно сохранить максимальный элемент из каждой группы, сравнить соответствующие произведения и выбрать из них наибольшее.

Сами числа при этом можно не хранить, достаточно при вводе определить остаток от деления очередного числа на 2 и на 5, сравнить число с текущим максимумом соответствующей группы и при необходимости обновить этот максимум и увеличить соответствующий счётчик. Таким образом, независимо от количества чисел в исходном наборе, после чтения исходных данных для хранения необходимой информации хватит 4 переменных, и программа получится эффективной по памяти.

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}
  m1: integer;     {нечётные, не кратные 5}
  m2: integer;     {чётные, не кратные 5}
  m5: integer;     {нечётные, кратные 5}
  m10: integer;    {чётные, кратные 5}
  x, y: integer;   {выбранная пара}
  i: integer;

begin
  m1:=0; m2:=0; m5:=0; m10:=0;
  readln(N);
  for i:=1 to N do begin
    readln(a);
    if a mod 2 = 0 then begin
      if a mod 5 = 0 then begin
        if a>m10 then m10 := a
      end
    else begin
      if a>m2 then m2 := a
    end
  end
  else begin
    if a mod 5 = 0 then begin
      if a>m5 then m5 := a
    else begin
      if a>m1 then m1 := a
    end
  end
  end;
  x:=m1; y:=m10;
  if m2*m5 > x*y then begin
    x:=m2; y:=m5;
  end
  if m5*m10 > x*y then begin
    x:=m5; y:=m10;
  end
  if x*y = 0
  then writeln(0)
  else writeln(x, ' ', y);
end.
```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары, выберем из них требуемую. 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}
    x, y: integer;   {выбранная пара}
    i, j: integer;

begin
    readln(N);
    for i:=1 to N do readln(a[i]);
    x := 0; y := 0;
    for i := 1 to N-1 do begin
        for j := i+1 to N do begin
            if ((a[i]+a[j]) mod 2 = 1) and
                ((a[i]*a[j]) mod 5 = 0) and
                (a[i]*a[j] > x*y)
            then begin
                x:=a[i]; y:=a[j]
            end
        end
    end
    end;
    if x = 0
    then writeln(0)
    else writeln(x, ' ', y);
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 групповых максимумов; 2) использование деления вместо нахождения остатка (div вместо mod в Паскале) или наоборот; 3) ошибка при построении системы проверок, в результате которой группы некоторых чисел определяются неверно; 4) учитываются не все возможные комбинации, дающие требуемый результат (например, не учитываются пары, в которых оба элемента кратны 5); 5) неверно определяется пара с максимальным произведением; 6) неверная обработка ситуации, в которой в наборе нет требуемых пар (например, вывод в этой ситуации двух чисел, как минимум одно из которых равно нулю); 7) неверный вывод ответа (например, вывод произведения вместо сомножителей) 	3

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. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла	2
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