

### Критерии оценивания заданий с развёрнутым ответом

24

Факториалом натурального числа  $n$  (обозначается  $n!$ ) называется произведение всех натуральных чисел от 1 до  $n$ . Например,  $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ . Дано целое положительное число  $A$ . Необходимо найти ближайшее к  $A$  число, которое является точным факториалом какого-либо натурального числа. Если два точных факториала одинаково близки к  $A$ , нужно выбрать меньший из них. Например, для  $A = 15$  нужно получить ответ 6, а для  $A = 20$  – ответ 24. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

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

Бейсик	Python
<pre> DIM A, K, F AS INTEGER INPUT A K = 1 F = 1 WHILE F &lt; A     K = K + 1     F = F * K WEND IF F - A &gt;= F \ K - A THEN     F = F \ K END IF PRINT K END         </pre>	<pre> a = int(input()) k = 1 f = 1 while f &lt; a:     k += 1     f *= k if f - a &gt;= f // k - a:     f //= k print(k)         </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел a, k, f     ввод a     k := 1     f := 1     нц пока f &lt; a         k := k + 1         f := f * k     кц     если f-a &gt;= div(f,k)-a         то f := div(f,k)     все     вывод k кон         </pre>	<pre> var a, k, f: integer; begin     read(a);     k := 1;     f := 1;     while f &lt; a do begin         k := k + 1;         f := f * k;     end;     if f-a &gt;= f div k - a         then f := f div k;     writeln(k); end.         </pre>

**C++**

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

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе  $A = 5$ .
2. Приведите пример числа  $A$ , при котором программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

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

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

<b>Содержание верного ответа</b>
----------------------------------

(допускаются иные формулировки ответа, не искажающие его смысла)
--

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. При вводе <math>A = 5</math> программа выведет число 3.</li> <li>2. Программа выводит верный ответ при вводе <math>A = 1</math> (ответ 1) или <math>A = 2</math> (ответ 2).</li> <li>3. Программа содержит две ошибки.</li> </ol> |
|---|

**Первая ошибка.** Ошибочная проверка. После завершения цикла гарантированно выполняется неравенство  $F \geq A > F/A$ , поэтому в сравнении после цикла левая часть всегда неотрицательна, а правая – отрицательна. Сравнение оказывается всегда верным, в результате выбирается не ближайший к  $A$  точный факториал, а ближайший, не превосходящий  $A$ .

**Вторая ошибка.** Ошибочный вывод. Вместо значения факториала  $F$  выводится «основание» факториала  $K$ .

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

```
if f-a >= f div k - a
```

Исправленная строка:

```
if f-a >= a - f div k
```

Возможно также использование функции `abs` (модуль числа):

```
if f-a >= abs(f div k - a)
```

Функцию `abs` можно использовать и в левой части сравнения. Это ничего не меняет по сути, так как левая часть заведомо неотрицательна, но такое использование нельзя считать ошибкой:

```
if abs(f-a) >= abs(f div k - a)
```

При использовании модуля для исправлении фрагмента на алгоритмическом языке следует использовать функцию `iabs` (модуль для целых чисел):

```
если f-a >= iabs(div(f,k)-a)
```

Использование в алгоритмическом языке функции `abs` вместо `iabs` отмечается как ошибка, но не влияет на оценку работы.

**Вторая ошибка:**

```
writeln(k)
```

Исправленная строка:

```
writeln(f)
```

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 (например, слова `THEN` в Бейсике или точки с запятой в `C++`).

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать ответ программы при данном вводе.</li> <li>2. Указать пример входного числа, при котором программа выдаёт верный ответ, и ответ программы в этом случае.</li> <li>3. Исправить первую ошибку в программе.</li> <li>4. Исправить вторую ошибку в программе.</li> </ol> <p>Действие 1 считается выполненным, если указан верный результат работы программы при заданном входном значении.</p> <p>Действие 2 считается выполненным, если указаны входное значение и ответ, при выполнении программы с этим входным значением получается этот ответ и этот ответ совпадает с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Для действий 1 и 2 экзаменуемый не обязан объяснять, как получен результат, достаточно указать верные числа.</p> <p>Каждое из действий 3 и 4 считается выполненным при одновременном выполнении двух условий:</p>	

а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).	
Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной.	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. 1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной. 2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной.	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий.	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
<i>Максимальный балл</i>	3

25

Дан массив, содержащий 2018 положительных целых чисел, не превышающих 15000. Необходимо уменьшить на удвоенное значение минимального элемента массива те элементы, которые более чем вдвое превышают этот минимум, и вывести изменённый массив по одному элементу в строке.

Например, для исходного массива из 5 элементов 40 100 27 90 54 программа должна вывести числа 40 46 27 36 54 по одному числу в строке (минимальный элемент исходного массива равен 27, элементы, бóльшие 54, уменьшены на 54).

Напишите на одном из языков программирования программу для решения этой задачи.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

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
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	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.
<b>C++</b>	
<pre> #include &lt;iostream&gt; using namespace std; const int N=2018; int main(){     int a[N];     int i, k, m;     for (i=0; i&lt;N; ++i)         cin &gt;&gt; a[i];      ...     return 0; } </pre>	

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Задача решается в два прохода: на первом проходе определяется минимальный 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>

<b>Пример правильной программы на языке Паскаль</b>	
<pre> m := 15001; for i:=1 to N do begin   if a[i] &lt; m then m := a[i] end; for i:=1 to N do begin   if a[i] &gt; 2*m then a[i] := a[i] - 2*m;   writeln(a[i]) end; </pre>	
<p>При использовании языка Python первый проход можно записать в одну строку, используя функцию min.</p>	
<b>Пример правильной программы на языке Python</b>	
<pre> m = min(a) for i in range(0,n):   if a[i] &gt; 2*m:     a[i] -= 2*m   print(a[i]) </pre>	
<p>Использовать описанную выше возможность не обязательно, на языке Python допустимо описывать развёрнутый алгоритм решения, аналогичный приведённой выше программе на языке Паскаль.</p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</p> <p>Эффективность не имеет значения и не оценивается.</p> <p>Допускается запись программы на языке, не входящем в список языков из условия. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если выбранный язык программирования использует типизированные переменные, 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>Допускается изменение указанного в условии формата вывода, например вывод всех элементов массива в одну строку.</p>	
Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив.	2
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку):</p> <p>1) отсутствие инициализации или неверная инициализация минимума;</p>	1

2) неверное определение минимума (например, поиск максимума вместо минимума); 3) выход за границы массива; 4) исходный массив не изменяется; 5) изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться; 6) неверное изменение (например, уменьшение на минимум, а не на удвоенный минимум); 7) полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы); 8) используется переменная, не объявленная в разделе описания переменных; 9) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	
Не выполнены условия, позволяющие поставить 1 или 2 балла.	0
Максимальный балл	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один камень** или **увеличить количество камней в куче в два раза**. Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (12, 9), (6, 10), (6, 18). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 50. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 50 или больше камней.

В начальный момент в первой куче было 8 камней, во второй куче –  $S$  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 играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

### Задание 1.

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

б) Назовите минимальное значение  $S$ , при котором Ваня может выиграть первым ходом в случае неудачного первого хода Пети.

### Задание 2.

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

### Задание 3.

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

Для указанного значения  $S$  опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

### Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

#### Задание 1.

а) Петя может выиграть единственным способом (увеличив количество камней во второй куче в два раза), если  $S = 21, \dots, 33$ . При меньших значениях  $S$  за один ход нельзя получить 50 или более камней в двух кучах. При  $S \geq 34$  у Пети есть более одного выигрывающего хода (можно удвоить количество камней в любой куче).

б) Минимальное значение:  $S = 11$ . Петя может получить позицию (8, 22), в которой Ваня может выиграть ходом (8, 44). При меньших значениях  $S$  ни при каком ходе Пети Ваня не сможет выиграть первым ходом.

#### Задание 2.

Возможные значения  $S$ : 16, 20. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при  $S = 16$  Петя может получить позицию (16, 16), а при  $S = 20$  – позицию (9, 20).

В первом случае после хода Вани возникнет одна из позиций (17, 16), (32, 16), (16, 17), (16, 32), во втором случае – одна из позиций (10, 20), (18, 20), (9, 21), (9, 40). В любой из перечисленных позиций Петя может выиграть, удвоив количество камней в большей куче.

#### Задание 3.

Возможное значение  $S$ : 19. После первого хода Пети возможны позиции (9, 19), (16, 19), (8, 20), (8, 38). В позициях (16, 19) и (8, 38) Ваня может выиграть первым ходом, удвоив количество камней в любой куче. Из позиций (9, 19) и (8, 20) Ваня может получить позицию (9, 20), разобранную



в задании 2. Игрок, после хода которого возникла эта позиция (в данном случае – Ваня), выигрывает следующим ходом.

В таблице изображены возможные партии при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом. На рисунке эти же партии показаны в виде графа (оба способа изображения допустимы).

	Положения после очередных ходов			
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(8, 19) Всего 27	$(8 \cdot 2, 19) = (16, 19)$ Всего 35	$(16 \cdot 2, 19) = (32, 19)$ <b>Всего 51</b>		
	$(8, 19 \cdot 2) = (8, 38)$ Всего 46	$(8 \cdot 2, 38) = (16, 38)$ <b>Всего 54</b>		
	$(8 + 1, 19) = (9, 19)$ Всего 28	$(9, 19 + 1) = (9, 20)$ или $(8 + 1, 20) = (9, 20)$ Всего 29	$(9 + 1, 20) = (10, 20)$ Всего 30	$(10, 20 \cdot 2) = (10, 40)$ <b>Всего 50</b>
			$(9 \cdot 2, 20) = (18, 20)$ Всего 38	$(18, 20 \cdot 2) = (18, 40)$ <b>Всего 58</b>
	$(8, 19 + 1) = (8, 20)$ Всего 28		$(9, 20 + 1) = (9, 21)$ Всего 30	$(9, 21 \cdot 2) = (9, 42)$ <b>Всего 51</b>
			$(9, 20 \cdot 2) = (9, 40)$ Всего 49	$(9, 40 \cdot 2) = (9, 80)$ <b>Всего 89</b>

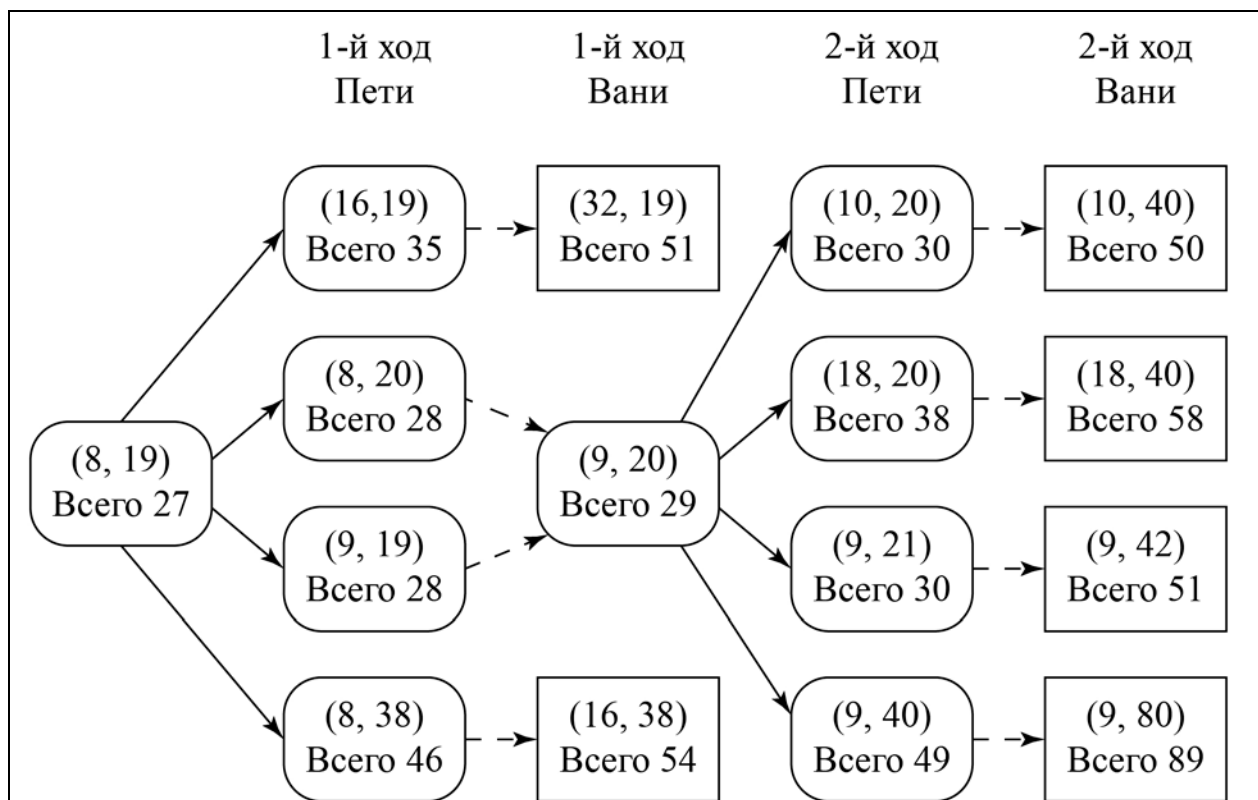


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани показаны пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*Примечание для эксперта.* Дерево всех партий может быть изображено в виде таблицы или в виде ориентированного графа – так, как показано на рисунке, или другим способом. Например, вместо приведённого здесь «экономного» варианта, в котором позиции не дублируются, возможно построение полного дерева, в котором одинаковые позиции, возникающие при различном ходе игры, показаны отдельно. Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три задания. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Задание 1 выполнено, если выполнены оба пункта: для пункта а) перечислены все удовлетворяющие условию значения <math>S</math> (и только они), для пункта б) указано верное значение <math>S</math> (и только оно). Обоснование найденных значений не обязательно.</p>	

Задание 2 выполнено, если верно указана выигрышная для Пети позиция (любая из двух возможных) и описана соответствующая стратегия.	
Задание 3 выполнено, если правильно указана выигрышная для Вани позиция и построено дерево всех возможных при выигрышной стратегии партий (и только их).	
Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.	
Выполнены все три задания.	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. <ul style="list-style-type: none"> <li>– Выполнено задание 3.</li> <li>– Выполнены задания 1 и 2.</li> </ul>	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из заданий 1 и 2.	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

27

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, находящихся на расстоянии не меньше 6 друг от друга (разница в индексах элементов должна быть 6 или более). Необходимо определить максимальную сумму такой пары.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $7 \leq N \leq 1000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10000.

*Пример входных данных:*

8  
1  
3  
5  
4  
6  
7  
9  
8

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

11

*Пояснение.* Из 8 чисел можно составить 3 пары, удовлетворяющие условию. Это будут элементы с индексами 1 и 7, 1 и 8, 2 и 8. Для заданного набора чисел получаем пары (1, 9), (1, 8), (3, 8). Максимальная сумма чисел в этих парах равна 11.

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.

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)
Будем рассматривать каждое введенное число как правый элемент возможной пары (первые 6 чисел не могут быть такими элементами). Для получения максимальной суммы нужно сложить это число с максимальным из всех элементов, расположенных от начала последовательности до элемента, расположенного на 6 позиций раньше текущего. Будем хранить этот максимум и корректировать его при вводе каждого нового элемента. Для этого понадобится хранить последние 6 элементов. Остальные элементы последовательности можно не хранить, это обеспечивает эффективность по памяти. Для хранения 6 элементов можно использовать циклический массив, как показано в следующем решении.

**Решение 1. Правильная и эффективная программы на языке Паскаль (использован циклический массив)**

```
const s=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[0..s-1] of integer;
  m: integer;      {максимальное число}
  sm: integer;     {максимальная сумма пары}
  i: integer;      {счётчик для ввода}
  ia: integer;     {текущий индекс в массиве a}
begin
  readln(N);
  {ввод первых s чисел}
  for i:=0 to s-1 do readln(a[i]);
  {ввод и обработка остальных значений}
  m:=0; sm:=0; ia:=0;
  for i:=s to N-1 do begin
    readln(x);
    if a[ia] > m then m := a[ia];
    if m+x > sm then sm := m+x;
    a[ia] := x;
    ia := (ia+1) mod s
  end;
  writeln(sm)
end.
```

Вместо циклического массива можно использовать сдвиги. В этом случае для вычисления максимума всегда используется первый элемент массива, а новое число записывается в последний. Хотя этот алгоритм работает медленнее, чем алгоритм с циклическим массивом (для каждого элемента требуется 5 дополнительных присваиваний при сдвигах), основное требование эффективности здесь выполнено: при увеличении размера массива в  $k$  раз количество действий растёт не более чем в  $k$  раз. Ниже приводится пример такой программы.

**Решение 2. Правильная и эффективная программы на языке Паскаль (использован сдвиг массива)**

```
const s=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[1..s] of integer;
  m: integer;      {максимальное число}
  sm: integer;     {максимальная сумма пары}
  i: integer;      {счётчик для ввода}
  ia: integer;     {счётчик для сдвига }
begin
  readln(N);
  {ввод первых s чисел}
  for i:=1 to s do readln(a[i]);
  {ввод и обработка остальных значений}
  m:=0; sm:=0;
  for i:=s+1 to N do begin
    readln(x);
    if a[1] > m then m := a[1];
    if m+x > sm then sm := m+x;
    for ia:=1 to s-1 do a[ia]:=a[ia+1];
    a[s] := x
  end;
  writeln(sm)
end.
```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и 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).

**Решение 3. Правильная, но неэффективная программы на языке Паскаль**

```

const s=6; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  a: array [1..1000] of integer; {исходные данные}
  sm: integer;     {максимальная сумма пары}
  i,j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  sm :=0;
  for i := 1 to N-s do begin
    for j := i+s to N do begin
      if a[i]+a[j] > sm
        then sm := a[i]+a[j]
    end
  end;
  writeln(sm)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается в 0 баллов.	
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, 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"> <li>1) поставлен лишний, пропущен или неверно указан знак пунктуации;</li> <li>2) написано лишнее, пропущено или неверно написано служебное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку.</p>	4
<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел, правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p>	3

<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"><li>1) ошибка при вводе данных (не считывается значение <math>N</math> или неверно организован ввод последовательности);</li><li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li><li>3) используется неверный тип данных;</li><li>4) использована одна переменная (константа) вместо другой;</li><li>5) используется один знак операции вместо другого;</li><li>6) отсутствует вывод ответа или выводится не то значение (хотя правильный ответ в программе найден);</li><li>7) неверная работа с массивом, в том числе выход за границы массива;</li><li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками).</li></ol>	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.</p> <p>2 балла также ставится за корректные переборные решения, в которых все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например выход индексов за границы массива, неверный учёт расстояния между элементами и т. д.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа представлена и содержит как минимум два обязательных элемента, возможно, реализованных с ошибками:</p> <ol style="list-style-type: none"><li>1) рассматриваются только пары, находящиеся на расстоянии не меньше заданного в условии;</li><li>2) сумма элементов пары сравнивается с максимумом.</li></ol>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла.</p>	0
<p>Максимальный балл</p>	4



### Критерии оценивания заданий с развёрнутым ответом

24

Факториалом натурального числа  $n$  (обозначается  $n!$ ) называется произведение всех натуральных чисел от 1 до  $n$ . Например,  $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ . Дано целое положительное число  $A$ . Необходимо найти ближайшее к  $A$  число, которое является точным факториалом какого-либо натурального числа. Если два точных факториала одинаково близки к  $A$ , нужно выбрать меньший из них. Например, для  $A = 15$  нужно получить ответ 6, а для  $A = 20$  – ответ 24. Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

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

Бейсик	Python
<pre> DIM A, K, F AS INTEGER INPUT A K = 0 F = 1 WHILE F &lt;= A     K = K + 1     F = F * K WEND IF F - A &gt; A - F \ K THEN     F = F \ K     K = K - 1 END IF PRINT K END </pre>	<pre> a = int(input()) k = 0 f = 1 while f &lt;= a:     k += 1     f *= k if f - a &gt; a - f//k:     f //= k     k -= 1 print(k) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач     цел a, k, f     ввод a     k := 0     f := 1     нц пока f &lt;= a         k := k + 1         f := f * k     кц     если f-a &gt; a - div(f,k)         то f := div(f,k)         k := k - 1     все     вывод k кон </pre>	<pre> var a, k, f: integer; begin     read(a);     k := 0;     f := 1;     while f &lt;= a do begin         k := k + 1;         f := f * k     end;     if f-a &gt; a - f div k     then begin         f := f div k;         k := k - 1     end;     writeln(k) end. </pre>

**C++**

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

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе  $A = 7$ .
2. Приведите пример числа  $A$ , при котором программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

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

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

Содержание верного ответа
---------------------------

(допускаются иные формулировки ответа, не искажающие его смысла)
--

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. При вводе <math>A = 7</math> программа выведет число 3.</li> <li>2. Программа выводит верный ответ при вводе <math>A = 1</math> (ответ 1), <math>A = 2</math> (ответ 2) и <math>A = 3</math> (ответ 2).</li> <li>3. Программа содержит две ошибки.</li> </ol> |
|---|

**Первая ошибка.** Ошибочная проверка. По условию при равном удалении  $A$  от двух точных факториалов нужно выбрать меньший из них. Для этого нужно строгое сравнение заменить на нестрогое.

**Вторая ошибка.** Ошибочный вывод. Вместо значения факториала  $F$  выводится «основание» факториала  $K$ .

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

```
if f-a > a - f div k
```

Исправленная строка:

```
if f-a >= a - f div k
```

Допускается использование функции `abs` (модуль числа). Это ничего не меняет по сути, так как обе части сравнения часть заведомо неотрицательны, но такое использование нельзя считать ошибкой:

```
if abs(f-a) >= abs(a - f div k)
```

При использовании функции модуля на алгоритмическом языке следует использовать функцию `iabs` (модуль для целых чисел):

```
если iabs(f-a) >= iabs(div(f,k)-a)
```

Использование в алгоритмическом языке функции `abs` вместо `iabs` отмечается как ошибка, но не влияет на оценку работы.

**Вторая ошибка:**

```
writeln(k)
```

Исправленная строка:

```
writeln(f)
```

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 (например, слова `THEN` в Бейсике или точки с запятой в C++).

Указания по оцениванию	Баллы
<p>В задаче требуется выполнить <b>четыре</b> действия.</p> <ol style="list-style-type: none"> <li>1. Указать ответ программы при данном вводе.</li> <li>2. Указать пример входного числа, при котором программа выдаёт верный ответ, и ответ программы в этом случае.</li> <li>3. Исправить первую ошибку в программе.</li> <li>4. Исправить вторую ошибку в программе.</li> </ol> <p>Действие 1 считается выполненным, если указан верный результат работы программы при заданном входном значении.</p> <p>Действие 2 считается выполненным, если указаны входное значение и ответ, при выполнении программы с этим входным значением получается этот ответ и этот ответ совпадает с ответом, который выдаёт для данного входного значения правильная программа.</p> <p>Для действий 1 и 2 экзаменуемый не обязан объяснять, как получен результат, достаточно указать верные числа.</p> <p>Каждое из действий 3 и 4 считается выполненным при одновременном выполнении двух условий:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</li> </ol>	

В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка).	
Выполнены все четыре необходимых действия, ни одна верная строка не указана в качестве ошибочной.	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций. <ol style="list-style-type: none"> <li>1. Выполнены три из четырёх необходимых действий, ни одна верная строка не названа ошибочной.</li> <li>2. Выполнены все четыре необходимых действия, одна верная строка названа ошибочной.</li> </ol>	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий.	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
<i>Максимальный балл</i>	3

**25** Дан массив, содержащий 2018 положительных целых чисел, не превышающих 15 000. Необходимо удвоить те элементы этого массива, которые меньше удвоенного минимального элемента, и вывести изменённый массив по одному элементу в строке.

Например, для исходного массива из 5 элементов 60 47 27 95 54 программа должна вывести числа 60 94 54 95 54 по одному числу в строке (минимальный элемент исходного массива равен 27, удвоены все элементы, меньшие 54).

Напишите на одном из языков программирования программу для решения этой задачи.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

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
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	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.
<b>C++</b>	
<pre>#include &lt;iostream&gt; using namespace std; const int N=2018; int main(){     int a[N];     int i, k, m;     for (i=0; i&lt;N; ++i)         cin &gt;&gt; a[i];      ...     return 0; }</pre>	

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
Задача решается в два прохода: на первом проходе определяется минимальный 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 := 15001; for i:=1 to N do begin     if a[i] &lt; m then m := a[i] end; for i:=1 to N do begin     if a[i] &lt; 2*m then a[i] := 2*a[i];     writeln(a[i]) end;</pre>
При использовании языка Python первый проход можно записать в одну строку, используя функцию min.

<b>Пример правильной программы на языке Python</b>	
<pre> m = min(a) for i in range(0,n):     if a[i] &lt; 2*m:         a[i] *= 2 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.	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>В программе допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора.</p> <p>Эффективность не имеет значения и не оценивается.</p> <p>Допускается запись программы на языке, не входящем в список языков из условия. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если выбранный язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на языках, использованных в задании. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи.</p> <p>Допускается изменение указанного в условии формата вывода, например вывод всех элементов массива в одну строку.</p>	
Предложена правильная программа, которая изменяет исходный массив в соответствии с условием и выводит изменённый массив.	2
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих (если одинаковая ошибка повторяется несколько раз, она считается за одну ошибку):</p> <ol style="list-style-type: none"> <li>1) отсутствие инициализации или неверная инициализация минимума;</li> <li>2) неверное определение минимума (например, поиск максимума вместо минимума);</li> <li>3) выход за границы массива;</li> <li>4) исходный массив не изменяется;</li> <li>5) изменяются не все элементы, которые должны измениться, или изменяются элементы, которые не должны измениться;</li> <li>6) неверное изменение;</li> <li>7) полученный массив не выводится или выводится не полностью (например, выводится только первый элемент или только изменённые элементы);</li> </ol>	1

8) используется переменная, не объявленная в разделе описания переменных;	
9) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.	
Не выполнены условия, позволяющие поставить 1 или 2 балла.	0
Максимальный балл	2

26

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один камень** или **увеличить количество камней в куче в два раза**. Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (12, 9), (6, 10), (6, 18). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 62. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 62 или больше камней.

В начальный момент в первой куче было 10 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 51$ .

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

### Задание 1.

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

б) Назовите минимальное значение  $S$ , при котором Ваня может выиграть первым ходом в случае неудачного первого хода Пети.

### Задание 2.

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

**Задание 3.**

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

Для указанного значения  $S$  опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

<b>Содержание верного ответа</b>
(допускаются иные формулировки ответа, не искажающие его смысла)
<p><b>Задание 1.</b></p> <p>а) Петя может выиграть единственным способом (увеличив количество камней во второй куче в два раза), если <math>S = 26, \dots, 41</math>. При меньших значениях <math>S</math> за один ход нельзя получить 62 или более камней в двух кучах. При <math>S \geq 42</math> у Пети есть более одного выигрывающего хода (можно удвоить количество камней в любой куче).</p> <p>б) Минимальное значение: <math>S = 13</math>. Петя может получить позицию (10, 26), в которой Ваня может выиграть ходом (10, 52). При меньших значениях <math>S</math> ни при каком ходе Пети Ваня не сможет выиграть первым ходом.</p> <p><b>Задание 2.</b></p> <p>Возможные значения <math>S</math>: 20, 25. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при <math>S = 20</math> Петя может получить позицию (20, 20), а при <math>S = 25</math> – позицию (11, 25).</p> <p>В первом случае после хода Вани возникнет одна из позиций (21, 20), (40, 20), (20, 21), (20, 40), во втором случае – одна из позиций (12, 25), (22, 25), (11, 26), (11, 50). В любой из перечисленных позиций Петя может выиграть, удвоив количество камней в большей куче.</p> <p><b>Задание 3.</b></p> <p>Возможное значение <math>S</math>: 24. После первого хода Пети возможны позиции (11, 24), (20, 24), (10, 25), (10, 48). В позициях (20, 24) и (10, 48) Ваня может выиграть первым ходом, удвоив количество камней в любой куче. Из позиций (11, 24) и (10, 25) Ваня может получить позицию (11, 25), разобрannую в задании 2. Игрок, после хода которого возникла эта позиция (в данном случае – Ваня), выигрывает следующим ходом.</p> <p>В таблице изображены возможные партии при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом. На рисунке эти же партии показаны в виде графа (оба способа изображения допустимы).</p>



	Положения после очередных ходов			
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(10, 24) Всего 34	$(10 \cdot 2, 24) = (20, 24)$ Всего 44	$(20 \cdot 2, 24) = (40, 24)$ <b>Всего 64</b>		
	$(10, 24 \cdot 2) = (10, 48)$ Всего 58	$(10 \cdot 2, 48) = (20, 48)$ <b>Всего 68</b>		
	$(10 + 1, 24) = (11, 24)$ Всего 35	$(11, 24 + 1) = (11, 25)$ или $(10 + 1, 25) = (11, 25)$ Всего 36	$(11 + 1, 25) = (12, 25)$ Всего 37	$(12, 25 \cdot 2) = (12, 50)$ <b>Всего 62</b>
			$(11 \cdot 2, 25) = (22, 25)$ Всего 47	$(22, 25 \cdot 2) = (22, 50)$ <b>Всего 72</b>
	$(10, 24 + 1) = (10, 25)$ Всего 35		$(11, 25 + 1) = (11, 26)$ Всего 37	$(11 \cdot 26 \cdot 2) = (11, 52)$ <b>Всего 63</b>
			$(11, 25 \cdot 2) = (11, 50)$ Всего 61	$(11, 50 \cdot 2) = (11, 100)$ <b>Всего 111</b>

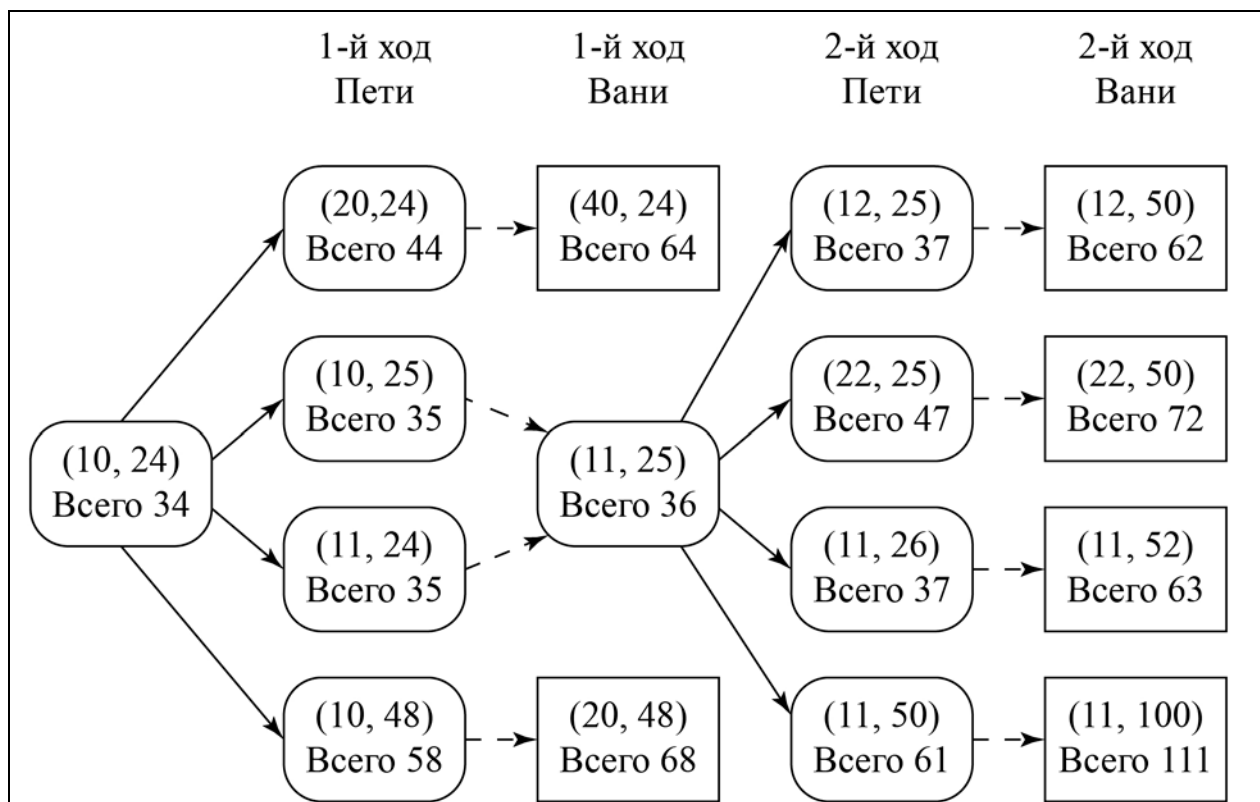


Рис. 1. Граф всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны сплошными стрелками, ходы Вани показаны пунктирными стрелками. Заключительные позиции обозначены прямоугольниками.

*Примечание для эксперта.* Дерево всех партий может быть изображено в виде таблицы или в виде ориентированного графа – так, как показано на рисунке, или другим способом. Например, вместо приведённого здесь «экономного» варианта, в котором позиции не дублируются, возможно построение полного дерева, в котором одинаковые позиции, возникающие при различном ходе игры, показаны отдельно. Важно, чтобы множество полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три задания. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>Задание 1 выполнено, если выполнены оба пункта: для пункта а) перечислены все удовлетворяющие условию значения <math>S</math> (и только они), для пункта б) указано верное значение <math>S</math> (и только оно). Обоснование найденных значений не обязательно.</p>	

Задание 2 выполнено, если верно указана выигрышная для Пети позиция (любая из двух возможных) и описана соответствующая стратегия. Задание 3 выполнено, если правильно указана выигрышная для Вани позиция и построено дерево всех возможных при выигрышной стратегии партий (и только их). Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом.	
Выполнены все три задания.	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. – Выполнено задание 3. – Выполнены задания 1 и 2.	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из заданий 1 и 2.	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла.	0
<i>Максимальный балл</i>	3

27

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, находящихся на расстоянии не меньше 8 друг от друга (разница в индексах элементов должна быть 8 или более). Необходимо определить максимальную сумму такой пары.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $9 \leq N \leq 1000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10000.

*Пример входных данных:*

10  
1  
3  
5  
4  
6  
7  
9  
10  
12  
11

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

14

*Пояснение.* Из 10 чисел можно составить 3 пары, удовлетворяющие условию. Это будут элементы с индексами 1 и 9, 1 и 10, 2 и 10. Для заданного набора чисел получаем пары (1, 12), (1, 11), (3, 11). Максимальная сумма чисел в этих парах равна 14.

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.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
Будем рассматривать каждое введенное число как правый элемент возможной пары (первые 8 чисел не могут быть такими элементами). Для получения максимальной суммы нужно сложить это число с максимальным из всех элементов, расположенных от начала последовательности до элемента, расположенного на 8 позиций раньше текущего. Будем хранить этот максимум и корректировать его при вводе каждого нового элемента. Для этого понадобится хранить последние 8 элементов. Остальные элементы последовательности можно не хранить, это обеспечивает эффективность по памяти. Для хранения 8 элементов можно использовать циклический массив, как показано в следующем решении.

**Решение 1. Правильная и эффективная программы на языке Паскаль (использован циклический массив)**

```
const s=8; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[0..s-1] of integer;
  m: integer;      {максимальное число}
  sm: integer;     {максимальная сумма пары}
  i: integer;      {счётчик для ввода}
  ia: integer;     {текущий индекс в массиве a}
begin
  readln(N);
  {ввод первых s чисел}
  for i:=0 to s-1 do readln(a[i]);
  {ввод и обработка остальных значений}
  m:=0; sm:=0; ia:=0;
  for i:=s to N-1 do begin
    readln(x);
    if a[ia] > m then m := a[ia];
    if m+x > sm then sm := m+x;
    a[ia] := x;
    ia := (ia+1) mod s
  end;
  writeln(sm)
end.
```

Вместо циклического массива можно использовать сдвиги. В этом случае для вычисления максимума всегда используется первый элемент массива, а новое число записывается в последний. Хотя этот алгоритм работает медленнее, чем алгоритм с циклическим массивом (для каждого элемента требуется 7 дополнительных присваиваний при сдвигах), основное требование эффективности здесь выполнено: при увеличении размера массива в  $k$  раз количество действий растёт не более чем в  $k$  раз. Ниже приводится пример такой программы.

**Решение 2. Правильная и эффективная программы на языке Паскаль (использован сдвиг массива)**

```
const s=8; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  x: integer;      {очередное число}
  a: array[1..s] of integer;
  m: integer;      {максимальное число}
  sm: integer;     {максимальная сумма пары}
  i: integer;      {счётчик для ввода}
  ia: integer;     {счётчик для сдвига }
begin
  readln(N);
  {ввод первых s чисел}
  for i:=1 to s do readln(a[i]);
  {ввод и обработка остальных значений}
  m:=0; sm:=0;
  for i:=s+1 to N do begin
    readln(x);
    if a[1] > m then m := a[1];
    if m+x > sm then sm := m+x;
    for ia:=1 to s-1 do a[ia]:=a[ia+1];
    a[s] := x
  end;
  writeln(sm)
end.
```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и 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).

**Решение 3. Правильная, но неэффективная программы на языке Паскаль**

```

const s=8; {требуемое расстояние между элементами}
var
  N: integer;      {количество чисел}
  a: array [1..1000] of integer; {исходные данные}
  sm: integer;     {максимальная сумма пары}
  i,j: integer;

begin
  readln(N);
  for i:=1 to N do readln(a[i]);
  sm :=0;
  for i := 1 to N-s do begin
    for j := i+s to N do begin
      if a[i]+a[j] > sm
        then sm := a[i]+a[j]
    end
  end;
  writeln(sm)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается в 0 баллов.	
<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, 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"> <li>1) поставлен лишний, пропущен или неверно указан знак пунктуации;</li> <li>2) написано лишнее, пропущено или неверно написано служебное слово языка программирования;</li> <li>3) не описана или неверно описана переменная;</li> <li>4) применяется операция, недопустимая для соответствующего типа данных.</li> </ol> <p>Если одна и та же ошибка встречается несколько раз, это считается за одну ошибку.</p>	4
<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел, правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.</p>	3

<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"> <li>1) ошибка при вводе данных (не считывается значение <math>N</math> или неверно организован ввод последовательности);</li> <li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li> <li>3) используется неверный тип данных;</li> <li>4) использована одна переменная (константа) вместо другой;</li> <li>5) используется один знак операции вместо другого;</li> <li>6) отсутствует вывод ответа или выводится не то значение (хотя правильный ответ в программе найден);</li> <li>7) неверная работа с массивом, в том числе выход за границы массива;</li> <li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками).</li> </ol>	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.</p> <p>2 балла также ставится за корректные переборные решения, в которых все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например выход индексов за границы массива, неверный учёт расстояния между элементами и т. д.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа представлена и содержит как минимум два обязательных элемента, возможно, реализованных с ошибками:</p> <ol style="list-style-type: none"> <li>1) рассматриваются только пары, находящиеся на расстоянии не меньше заданного в условии;</li> <li>2) сумма элементов пары сравнивается с максимумом.</li> </ol>	1
<p>Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла.</p>	0
<p>Максимальный балл</p>	4