

ФЕДЕРАЛЬНАЯ СЛУЖБА ПО НАДЗОРУ В СФЕРЕ ОБРАЗОВАНИЯ И НАУКИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ НАУЧНОЕ УЧРЕЖДЕНИЕ
«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»**

**Методические материалы для председателей
и членов региональных предметных комиссий
по проверке выполнения заданий с развернутым ответом
экзаменационных работ ЕГЭ 2016 года**

ИНФОРМАТИКА И ИКТ

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО
ОЦЕНИВАНИЮ ВЫПОЛНЕНИЯ ЗАДАНИЙ
ЕГЭ С РАЗВЕРНУТЫМ ОТВЕТОМ**

Москва
2016

Методические материалы для председателей и членов региональных предметных комиссий по проверке выполнения заданий с развернутым ответом экзаменационных работ ЕГЭ 2016 г. по информатике и ИКТ подготовлены в соответствии с Тематическим планом работ Федерального государственного бюджетного научного учреждения «Федеральный институт педагогических измерений» на 2016 г. Пособие предназначено для подготовки экспертов по оцениванию заданий с развернутым ответом, которые являются частью контрольных измерительных материалов (КИМ) для сдачи единого государственного экзамена (ЕГЭ) по информатике и ИКТ.

В методических материалах дается краткое описание структуры контрольных измерительных материалов 2016 г. по информатике и ИКТ, характеризуются типы заданий с развернутым ответом, используемые в КИМ ЕГЭ по информатике и ИКТ, и критерии оценки выполнения заданий с развернутым ответом, приводятся примеры оценивания выполнения заданий и даются комментарии, объясняющие выставленную оценку.

Авторы будут благодарны за замечания и предложения по совершенствованию пособия.

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОДХОДЫ К РАЗРАБОТКЕ КОНТРОЛЬНЫХ ИЗМЕРИТЕЛЬНЫХ МАТЕРИАЛОВ ЕГЭ ПО ИНФОРМАТИКЕ И ИКТ	4
2. ХАРАКТЕРИСТИКА РАЗНЫХ ТИПОВ ЗАДАНИЙ С РАЗВЕРНУТЫМ ОТВЕТОМ И РЕКОМЕНДАЦИИ ПО ИХ ОЦЕНИВАНИЮ	5
2.1. ЗАДАНИЯ С РАЗВЕРНУТЫМ ОТВЕТОМ В СТРУКТУРЕ КОНТРОЛЬНЫХ ИЗМЕРИТЕЛЬНЫХ МАТЕРИАЛОВ ДЛЯ ЕГЭ ПО ИНФОРМАТИКЕ И ИКТ	5
Фрагменты спецификации экзаменационной работы по информатике 2016 года, относящиеся к заданиям части 2	5
Распределение заданий с развернутым ответом по уровню сложности	5
Система оценивания выполнения заданий с развернутым ответом и экзаменационной работы в целом	5
Фрагмент обобщенного плана экзаменационной работы по информатике и ИКТ 2016 г. ..	5
2.2. ВАРИАНТЫ ЗАДАНИЙ ЧАСТИ 2 И КРИТЕРИИ ОЦЕНИВАНИЯ....	7
Варианты задания 24 и критерии оценивания	7
Задание 24 . Вариант 1.	7
Задание 24. Вариант 2	9
Задание 24. Вариант 1а.	10
Задание 24. Вариант 2а.	13
Варианты задания 25 и критерии оценивания	16
Задание 25. Вариант 1.	16
Задание 25. Вариант 2.	18
Варианты задания 26 и критерии оценивания	21
Задание 26. Вариант 1.	21
Задание 26. Вариант 2.	23
Вариант задания 27 и критерии оценивания	27
Задание 27. Вариант 1.	27
ИНСТРУКЦИИ И ПАМЯТКИ ПО ПРОВЕРКЕ РАБОТ	35
Задание 24	35
Задание 25	36
Задание 26	36
Задание 27	36
ОПИСАНИЕ СИТУАЦИЙ СЛОЖНЫХ ДЛЯ ОЦЕНИВАНИЯ	38
Задание 24	38
Задание 25	38
Задание 26	39
Задание 27	39

1. Общие подходы к разработке контрольных измерительных материалов ЕГЭ по информатике и ИКТ

Разработка системы единого государственного экзамена включает в себя создание большого количества взаимосвязанных подсистем. Одной из них является формирование комплекса стандартизированной подготовки экспертов-предметников, включающей эффективное обучение проверке заданий с развернутыми ответами контрольных измерительных материалов (в частности, по информатике) с точным соблюдением централизованно разработанных критериев оценивания выполнения учащимися заданий с развернутыми ответами. Решение этой задачи – одно из условий обеспечения объективности и надежности результатов, полученных в ходе единого государственного экзамена.

Предлагаемые учебно-методические материалы (УММ) для подготовки экспертов, привлекаемых для проверки заданий с развернутыми ответами по информатике и ИКТ в рамках ЕГЭ, разработаны на основе открытых вариантов КИМ ЕГЭ и анализа опыта подготовки экспертов последних лет. Предлагаемые материалы учитывают спецификацию экзаменационной работы 2016 года.

Экзаменационная работа состоит из двух частей. Часть 1 содержит 23 задания с кратким ответом по всем основным разделам курса информатики. Задания части 2 направлены на проверку сформированности важнейших умений записи и анализа алгоритмов, предусмотренных требованиями к обязательному уровню подготовки по информатике учащихся общеобразовательных учреждений. В этой части также проверяются умения выпускников решать задачи на повышенном и высоком уровне сложности по теме «Технология программирования». Решения заданий части 2 работы записываются в развернутой форме и проверяются экспертами региональных предметных комиссий. За выполнение каждого задания дается определенное количество баллов, в зависимости от полноты и качества выполнения. Так, часть 2 включает 4 задания, что составляет почти 15% от общего количества заданий. При успешном их выполнении экзаменуемый может получить максимально 12 первичных баллов (т. е. примерно треть общего количества первичных баллов за всю работу). С другой стороны, эти задания являются самыми сложными и самыми трудоемкими: рекомендованное время их выполнения в 1,6 раза превосходит время, отводимое на выполнение первой части работы.

2. Характеристика разных типов заданий с развернутым ответом и рекомендации по их оцениванию

2.1. Задания с развернутым ответом в структуре контрольных измерительных материалов для ЕГЭ по информатике и ИКТ

Фрагменты спецификации экзаменационной работы по информатике 2016 года, относящиеся к заданиям части 2

Задания части 2 направлены на проверку сформированности важнейших умений записи и анализа алгоритмов, предусмотренных требованиями к обязательному уровню подготовки по информатике учащихся средних общеобразовательных учреждений. Эти умения проверяются на повышенном и высоком уровне сложности. Также на высоком уровне сложности проверяются умения по теме «Технология программирования».

Распределение заданий с развернутым ответом по уровню сложности

В части 2 всего четыре задания, относящиеся к повышенному и высокому уровню сложности.

Если для заданий базового уровня предполагаемый процент выполнения 60%–80%, то для заданий повышенного и высокого уровня сложности требования более высокие. Для задания 24 повышенного уровня предполагаемый процент выполнения от 40 % до 60%, а для остальных заданий части 2 предполагаемый процент выполнения от 10% до 30%.

Система оценивания выполнения заданий с развернутым ответом и экзаменационной работы в целом

Ответы на задания части 2 проверяются и оцениваются экспертами (устанавливается соответствие ответов определенному перечню критериев).

Максимальное количество баллов, которое можно получить за выполнение заданий части 2, – 12 баллов.

Фрагмент обобщенного плана экзаменационной работы по информатике и ИКТ 2016 г.

№ п/п	Обозначение задания	Проверяемые элементы содержания и виды деятельности	Коды проверяемых элементов содержания по кодификатору	Коды требований к уровню подготовки выпускников по кодификатору	Уровень сложности задания	Максимальный балл за задание	Примерное время выполнения (мин)
...	...	...	...	...	...	...	...
		Часть 3					
24	24	Умение прочесть фрагмент программы на языке программирования и исправить допущенные ошибки	1.7.2	1.1.4	П	3	30
25	25	Умения написать короткую (10–15 строк) простую программу	1.6.3	1.1.5	В	2	30

		обработки массива на языке программирования или записать алгоритм на естественном языке					
26	26	Умение построить дерево игры по заданному алгоритму и обосновать выигрышную стратегию	1.5.2	1.1.3	В	3	30
27	27	Умения создавать собственные программы (30–50 строк) для решения задач средней сложности	1.7.3	1.1.5	В	4	55

2.2. Варианты заданий части 2 и критерии оценивания

Варианты задания 24 и критерии оценивания

Задание 24 . Вариант 1.

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N , не превосходящее 10^9 , и выводится количество цифр этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках программирования.)

Бейсик	Паскаль
<pre>DIM N AS LONG INPUT N sum = 0 WHILE N >= 9 N = N \ 10 sum = sum + 1 WEND PRINT sum END</pre>	<pre>var N: longint; sum: integer; begin readln(N); sum := 0; while N >= 9 do begin N := N div 10; sum := sum + 1; end; writeln(sum); end.</pre>
Си	Алгоритмический язык
<pre>#include<stdio.h> int main() { long int N; int sum; scanf("%ld", &N); sum = 0; while (N >= 9) { N = N / 10; sum = sum + 1; } printf("%d", sum); }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N, sum <u>ввод</u> N sum := 0 <u>нц пока</u> N >= 9 N := div(N, 10) sum := sum + 1 <u>кц</u> <u>вывод</u> sum <u>кон</u></pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 584.
2. Укажите число, для которого программа будет работать верно.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Укажите все строки (одну или более), содержащие ошибки, и для каждой такой строки приведите правильный вариант.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
1. Программа выведет число 2. 2. Программа работает верно для всех чисел, начинающихся на 9, в том числе для числа 9. [Достаточно указать любое такое число.] 3. В качестве ответа для остальных чисел программа выдаёт число на 1 меньшее, чем нужно. Возможные (не все) варианты исправления для языка Паскаль: 1) исправление условия продолжения цикла на while (N >= 1) do или while (N > 0) do При этом замена на while (N >= 0) do корректной не является. 2) исправление инициализации на sum := 1 а условие продолжения цикла на while (N > 9) do или while (N >= 10) do	
Указания по оцениванию	Баллы
Обратите внимание! В задаче требовалось выполнить три действия. Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. 1. Верно указано, что именно выведет программа при указанных в условии входных данных. 2. Указано число, при котором программа работает верно. 3. Указаны все строки (одна или более), в которые нужно внести исправления, и эти исправления внесены; при этом получена верно работающая программа. При выполнении действия 3 верное указание на ошибку при неверном её исправлении не засчитывается. Обратите внимание! Выбор ошибочных строк может быть выполнен не единственным способом. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	
Правильно выполнены все три действия	3
Правильно выполнены два действия из трёх	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, однако выполнено одно из следующих условий. 1. Выполнено одно действие из трёх. 2. Представлен новый верный текст программы, возможно, совершенно не похожий на исходный	1
Все пункты задания выполнены неверно или отсутствуют	0
<i>Максимальный балл</i>	3

Задание 24. Вариант 2

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N , не превосходящее 10^9 , и выводится количество цифр этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках.)

Бейсик	Паскаль
<pre> DIM N AS LONG INPUT N sum = 1 WHILE N > 1 N = N \ 10 sum = sum + 1 WEND PRINT sum END </pre>	<pre> var N: longint; sum: integer; begin readln(N); sum := 1; while N > 1 do begin N := N div 10; sum := sum + 1; end; writeln(sum); end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int main() { long int N; int sum; scanf("%ld", &N); sum = 1; while (N > 1) { N = N /10; sum = sum + 1; } printf("%d", sum); } </pre>	<pre> алг нач цел N, sum <u>ввод</u> N sum := 1 <u>нц пока</u> N > 1 N := div(N, 10) sum := sum + 1 <u>кц</u> <u>вывод</u> sum кон </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 938?
2. Укажите одно число, для которого программа будет работать верно.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Укажите все строки (одну или более), содержащие ошибки, и для каждой такой строки приведите правильный вариант.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

1. Программа выведет число 4. 2. Программа работает верно для любого числа, начинающегося с 1, в том числе для 1. [Достаточно указать любое такое число.] 3. В качестве ответа для остальных чисел программа выдаёт число на 1 большее, чем нужно. Возможные варианты исправления для языка Паскаль: 1) исправление условия продолжения цикла на while (N > 9) do 2) исправление инициализации на sum := 0 а условие продолжения цикла на while (N >= 1) do или while (N > 0) do При этом замена на while (N >= 0) do корректной не является. 3) исправление условия продолжения цикла на while (N >= 1) do или while (N > 0) do и вывод значения sum-1	
Указания по оцениванию	Баллы
Обратите внимание! В задаче требовалось выполнить три действия. Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. 1. Верно указано, что именно выведет программа при указанных в условии входных данных. 2. Указано число, при котором программа работает верно. 3. Указаны все строки (одна или более), в которые нужно внести исправления, и эти исправления внесены; при этом получена верно работающая программа. При выполнении действия 3 верное указание на ошибку при неверном её исправлении не засчитывается. Обратите внимание! Выбор ошибочных строк может быть выполнен не единственным способом. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	
Правильно выполнены все три действия	3
Правильно выполнены два действия из трёх	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, однако выполнено одно из следующих условий. 1. Выполнено одно действие из трёх. 2. Представлен новый верный текст программы, возможно, совершенно не похожий на исходный	1
Все пункты задания выполнены неверно или отсутствуют	0
Максимальный балл	3

Задание 24. Вариант 1а.

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N , не превосходящее 10^9 , и выводится минимальная цифра этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках программирования.)

Бейсик	Паскаль
DIM N AS LONG	var N: longint;

<pre> INPUT N min_digit = 9 WHILE N >= 10 digit = N MOD 10 IF digit < min_digit THEN min_digit = digit END IF N = N \ 10 WEND PRINT digit END </pre>	<pre> digit, min_digit: integer; begin readln(N); min_digit := 9; while N >= 10 do begin digit := N mod 10; if digit < min_digit then min_digit := digit; N := N div 10; end; writeln(digit); end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int main() { long int N; int digit, min_digit; scanf("%ld", &N); min_digit = 9; while (N >= 10) { digit = N % 10; if (digit < min_digit) min_digit = digit; N = N / 10; } printf("%d", digit); } </pre>	<pre> алг нач цел N, digit, min_digit ввод N min_digit := 9 нц пока N >= 10 digit := mod(N, 10) если digit < min_digit то min_digit := digit все N := div(N, 10) кц вывод digit кон </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 547.
2. Приведите пример числа, при вводе которого программа работает правильно, несмотря на содержащиеся в ней ошибки.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
 - 1) выпишите строку, в которой сделана ошибка;
 - 2) укажите, как исправить ошибку, – приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
--

Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках. 1. Программа выведет число 4. 2. Верным ответом является любое число $10 \leq N \leq 10^9$, минимальной цифрой которого является вторая цифра слева. Например, число 11 или 9899. 3. В программе есть две ошибки. Первая ошибка. Неверное условие окончания цикла. Программа не будет рассматривать старшую цифру числа. Строка с ошибкой: <pre>while N >= 10 do</pre> Возможные варианты исправления: <pre>while (N >= 1)</pre> или <pre>while (N > 0)</pre> При этом замены на <pre>while (N > 1) или while (N >= 0)</pre> корректными не являются 3. Вторая ошибка. Программа выводит значение переменной <code>digit</code>, а не <code>min_digit</code>. В результате программа всегда выводит вторую слева цифру числа. Строка с ошибкой: <pre>writeln(digit);</pre> Необходимо в строке с выводом результата заменить <code>digit</code> на <code>min_digit</code>	
Указания по оцениванию	Баллы
Обратите внимание! В задаче требовалось выполнить четыре действия: 1) указать, что выведет программа при конкретных входных данных; 2) указать пример подаваемого на вход числа, для которого программа выведет верный результат; 3) исправить первую ошибку; 4) исправить вторую ошибку. Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы. Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.	
Выполнены все четыре 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: а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два	1

необходимых действия из четырёх	
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
Максимальный балл	3

Задание 24. Вариант 2а.

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N , не превосходящее 10^9 , и выводится минимальная цифра этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках программирования.)

Бейсик	Паскаль
<pre> DIM N AS LONG INPUT N min_digit = 0 WHILE N > 0 digit = N MOD 10 IF digit < min_digit THEN min_digit = digit END IF N = N \ 10 WEND PRINT digit END </pre>	<pre> var N: longint; digit, min_digit: integer; begin readln(N); min_digit := 0; while N > 0 do begin digit := N mod 10; if digit < min_digit then min_digit := digit; N := N div 10; end; writeln(digit); end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int main() { long int N; int digit, min_digit; scanf("%ld", &N); min_digit = 0; while (N > 0) { digit = N % 10; if (digit < min_digit) min_digit = digit; N = N / 10; } printf("%d", digit); } </pre>	<pre> алг нач цел N, digit, min_digit ввод N min_digit := 0 нц пока N > 0 digit := mod(N, 10) если digit < min_digit то min_digit := digit все N := div(N, 10) кц вывод digit кон </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 862.
2. Приведите пример числа, при вводе которого программа работает правильно, несмотря на содержащиеся в ней ошибки.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:
 - 1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, – приведите правильный вариант строки. Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках. 1. Программа выведет число 8. 2. Верным ответом является любое число $1 \leq N \leq 10^9$, минимальной цифрой которого является самая левая цифра. Например, числа 1, 22 или 789. 3. В программе есть две ошибки Первая ошибка. Неверная инициализация ответа (переменная <code>min_digit</code>). Строка с ошибкой: <pre>min_digit := 0;</pre> Возможные варианты исправления: <pre>min_digit := 9;</pre> Возможны и другие исправления инициализации на любое число, большее 9. Вторая ошибка. Программа выводит значение переменной <code>digit</code>, а не <code>min_digit</code>. В результате программа всегда выводит самую старшую цифру числа. Строка с ошибкой: <pre>writeln(digit);</pre> Необходимо в строке с выводом результата заменить <code>digit</code> на <code>min_digit</code>	
Указания по оцениванию	Баллы
Обратите внимание! В задаче требовалось выполнить четыре действия: 1) указать, что выведет программа при конкретных входных данных; 2) указать пример подаваемого на вход числа, для которого программа выведет верный результат; 3) исправить первую ошибку; 4) исправить вторую ошибку. Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы. Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.	
Выполнены все четыре 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: а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано	2

в качестве ошибочной не более одной верной строки	
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
<i>Максимальный балл</i>	<i>3</i>

Варианты задания 25 и критерии оценивания

Задание 25. Вариант 1.

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 150 до 200 – рост учащихся выпускного класса. В команду по автогонкам входят все учащиеся, чей рост не более 175 см. Гарантируется, что такие учащиеся в классе есть. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит рост самого высокого участника гоночной команды.

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

Паскаль	Бейсик
<pre>const N=30; var a: array [1..N] of integer; i, j, max: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=30 DIM A(N) AS INTEGER DIM I, J, MAX AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Алгоритмический язык
<pre>#include <stdio.h> #define N 30 void main(void) { int a[N]; int i, j, max; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 30 <u>целтаб</u> a[1:N] <u>цел</u> i, j, max <u>нц</u> для i от 1 до N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Русский (естественный) язык	
Объявляем массив А из 30 элементов. Объявляем целочисленные переменные I, J, MAX. В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й. ...	

В качестве ответа необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
На языке Паскаль	На языке Бейсик
<pre> max:=150; for i:=1 to N do if (a[i]<=175) and (a[i]>max) then max:=a[i]; writeln(max); </pre>	<pre> MAX = 150 FOR I = 1 TO N IF A(I) <= 175 AND A(I) > MAX THEN MAX = A(I) ENDIF NEXT I PRINT MAX </pre>
На языке СИ	На Алгоритмическом языке
<pre> max=150; for (i=0; i<N; i++) if (a[i]<=175 && a[i]>max) max=a[i]; printf("% d", max); </pre>	<pre> max = 150 нц для i от 1 до N если a[i]<=175 и a[i]> max то max := a[i] все кц вывод max </pre>
На русском (естественном) языке	
Записываем в переменную MAX начальное значение, равное 150. В цикле от первого элемента до тридцатого сравниваем элементы исходного массива со 175. Если текущий элемент меньше или равен 175, то сравниваем значение текущего элемента массива со значением переменной MAX. Если текущий элемент массива больше MAX, то записываем в MAX значение этого элемента массива. Переходим к следующему элементу. После завершения цикла выводим значение переменной MAX.	

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. 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 алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.	2
В любом варианте решения может присутствовать не более одной ошибки из числа следующих. 1. Не инициализируется или неверно инициализируется переменная MAX (например, ей присваивается значение a[1] или число большее 150). 2. В сравнении со 175 вместо знака «меньше или равно» используется знак «меньше». 3. Отсутствует вывод ответа. 4. Используется переменная, не объявленная в разделе описания переменных. 5. Не указано или неверно указано условие завершения цикла. 6. Индексная переменная в цикле не меняется (например, в цикле while). 7. Неверно расставлены операторные скобки.	1
Ошибок, перечисленных в п. 1–7, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

Задание 25. Вариант 2.

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое элементов массива, имеющих нечетное значение. Гарантируется, что в исходном массиве хотя бы один элемент имеет нечетное значение. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre>const N=30; var a: array [1..N] of integer; i, x, y: integer; s: real; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=30 DIM A(N) AS INTEGER DIM I, X, Y AS INTEGER DIM S AS SINGLE FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Алгоритмический язык
<pre>#include <stdio.h> #define N 30</pre>	<pre><u>алг</u> нач</pre>

<pre> void main(void) { int a[N]; int i, x, y; float s; for (i=0; i<N; i++) scanf("%d", &a[i]); ... } </pre>	<pre> <u>цел</u> N = 30 <u>целтаб</u> a[1:N] <u>цел</u> i, x, y <u>вещ</u> s <u>нц</u> для i от 1 до N ввод a[i] <u>кц</u> ... <u>кон</u> </pre>
---	--

Русский (естественный) язык

Объявляем массив А из 30 элементов.
Объявляем целочисленные переменные
I, X, Y.
Объявляем вещественную переменную S.
В цикле от 1 до 30 вводим элементы
массива А с 1-го по 30-й.
...

В качестве ответа необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
На языке Паскаль	На языке Бейсик
<pre> x:=0; y:=0; for i:=1 to N do if (a[i] mod 2=1) then begin x:=x+a[i]; y:=y+1; end; s:=x/y; writeln(s); </pre>	<pre> X = 0 Y = 0 FOR I = 1 TO N IF A(I) MOD 2 = 1 THEN X = X + A(I) Y = Y + 1 ENDIF NEXT I S = X / Y PRINT S </pre>
На языке СИ	На Алгоритмическом языке
<pre> x=0; y=0; for (i=0; i<N; i++) if (a[i]%2==1) { x=x+a[i]; y++; } s=(float)x/y; printf("%f", s); </pre>	<pre> x:=0 y:=0 <u>нц</u> для i от 1 до N <u>если</u> mod(a[i],2) = 1 <u>то</u> x := x+ a[i] y := y+1 <u>все</u> <u>кц</u> s := x/y <u>ВЫВОД</u> s </pre>
На русском (естественном) языке	

Записываем в переменные X и Y начальное значение, равное нулю. В цикле от первого элемента до тридцатого находим остаток от деления элемента исходного массива на два. Если этот остаток равен единице, то увеличиваем счетчик суммы X на значение текущего элемента массива, а счетчик количества Y на 1. Переходим к следующему элементу.

После цикла производим деление счетчика суммы X на счетчик количества Y и записываем результат в переменную S. Выводим значение переменной S.

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. 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 алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.	2
В любом варианте решения присутствует не более одной ошибки из числа следующих. 1. Значения переменных X и Y находятся верно, однако среднее арифметическое считается неверно (например, производится действие X/N или неверно происходит преобразование типов при делении). 2. Неверно осуществляется проверка значения элемента массива на четность. 3. Не инициализируются или неверно инициализируются переменные X и Y. 4. Отсутствует вывод ответа. 5. Используется переменная, не объявленная в разделе описания переменных. 6. Не указано или неверно указано условие завершения цикла. 7. Индексная переменная в цикле не меняется (например, в цикле while). 8. Неверно расставлены операторные скобки.	1
Ошибок, перечисленных в п. 1–7, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

Варианты задания 26 и критерии оценивания

Задание 26. Вариант 1.

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 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 22. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 22 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 21$.

Говорят, 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.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) При каких значениях числа S Петя может выиграть первым ходом? Укажите все такие значения.
 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.
2. Укажите два значения S , при которых у Пети есть выигрышная стратегия, причем (а) Петя не может выиграть первым ходом, но (б) Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.
 Для указанных значений S опишите выигрышную стратегию Пети.
3. Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но 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 ребрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
(допускаются иные формулировки ответа, не искажающие его смысла)	
1.	а) Петя может выиграть первым ходом, если $S = 11, \dots, 21$. Во всех случаях нужно удвоить количество камней в куче. При меньших значениях S за один ход нельзя получить кучу, в которой больше 21 камня. б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 10$ камней. Тогда после первого хода Пети в куче будет 11 камней или 20 камней. В обоих случаях Ваня удваивает количество камней и выигрывает первым ходом.
2.	Возможные значения S: 5 и 9. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 10 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть, Петя) следующим ходом выиграет.

3. Возможное значение S : 8. После первого хода Пети в куче будет 9 или 16 камней. Если в куче станет 16 камней, Ваня удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 9 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом. В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчеркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
8	$8+1=9$	$9+1=10$	$10+1=11$	$11*2=22$
			$10*2=20$	$20*2=40$
	$8*2=16$	$16*2=32$		

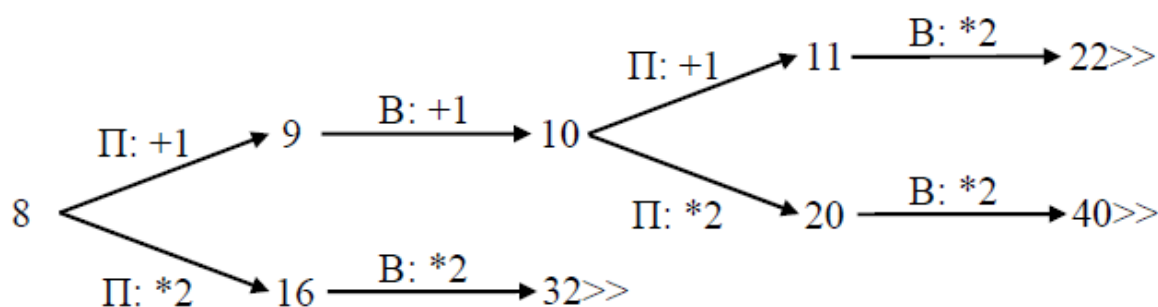


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже). Ошибка в решении, не искажающая основного замысла, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается. Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом и описана стратегия Вани, т.е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней при любом ходе Пети. Первое задание считается выполненным частично, если (1) правильно указаны все позиции, в которых Петя выигрывает первым ходом (пункт а), (2) правильно указана позиция, в которой Ваня выигрывает первым ходом; (3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для	

выигрыша количество камней. Второе задание выполнено, если правильно указаны обе позиции, выигрышная для Пети и описана соответствующая стратегия Пети – так, как это написано в примере решения или другим способом, например, с помощью дерева всех возможных партий. Третье задание выполнено, если правильно указана позиция, выигрышная для Вани и построено дерево всех партий, возможных при Ваниной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня – только ход, соответствующий стратегии, которую выбрал Ваня. Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения или другим способом.	
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше).	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Задание 3 выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S.	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое задание выполнено полностью. 2. Во втором задании правильно указано одно из двух возможных значений S, и для этого значения указана и обоснована выигрышная стратегия Пети. 3. Первое задание выполнено полностью или частично и для одного из остальных заданий правильно указано значение S. 4. Для второго и третьего задания правильно указаны значения S.	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл	0
Максимальный балл	
	3

Задание 26. Вариант 2.

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 1 камень или увеличить количество камней в куче в 3 раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 30. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 30 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 29$.

Говорят, 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.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

- а) При каких значениях числа S Петя может выиграть первым ходом? Укажите все такие значения.
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.
- Укажите два значения S , при которых у Пети есть выигрышная стратегия, причем (а) Петя не может выиграть первым ходом, но (б) Петя может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.

Для указанных значений S опишите выигрышную стратегию Пети.

- Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но 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 ребрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа и указания по оцениванию
(допускаются иные формулировки ответа, не искажающие его смысла)

- а) Петя может выиграть первым ходом, если $S = 10, \dots, 29$. Во всех случаях нужно утроить количество камней в куче. При меньших значениях S за один ход нельзя получить кучу, в которой 30 или больше камней.
б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 9$ камней. Тогда после первого хода Пети в куче будет 10 камней или 27 камней. В обоих случаях Ваня утраивает количество камней и выигрывает первым ходом.
- Возможные значения S : 3 и 8. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 9 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть, Петя) следующим ходом выигрывает.
- Возможное значение S : 7. После первого хода Пети камней в куче будет 8 или 21. Если в куче станет 21 камень, Ваня утроит количество камней и выигрывает первым ходом. Ситуация, когда в куче 8 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.
В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчеркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов				
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
7	$7+1=8$	$8+1=9$	$9+1=10$	<u>$10 \cdot 3=30$</u>

			$9 \cdot 3 = \mathbf{27}$	$27 \cdot 3 = \mathbf{81}$
	$7 \cdot 3 = \mathbf{21}$	$21 \cdot 3 = \mathbf{63}$		

Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже). Ошибка в решении, не искажающая основного замысла, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается. Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом и указано, каким должен быть первый ход. Пункт б) считается выполненным, если правильно указана позиция, в которой Ваня выигрывает первым ходом и описана стратегия Вани, т.е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней при любом ходе Пети. Первое задание считается выполненным частично, если (1) правильно указаны все позиции, в которых Петя выигрывает первым ходом (пункт а), (2) правильно указана позиция, в которой Ваня выигрывает первым ходом; (3) явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. Второе задание выполнено, если правильно указаны обе позиции, выигрышная для Пети и описана соответствующая стратегия Пети – так, как это написано в примере решения или другим способом, например, с помощью дерева всех возможных партий. Третье задание выполнено, если правильно указана позиция, выигрышная для Вани и построено дерево всех партий, возможных при Ваниной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня – только ход, соответствующий стратегии, которую выбрал Ваня. Во всех случаях стратегии могут быть описаны так, как это сделано в примере	

решения или другим способом.	
Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше).	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Задание 3 выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S .	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое задание выполнено полностью. 2. Во втором задании правильно указано одно из двух возможных значений S , и для этого значения указана и обоснована выигрышная стратегия Пети. 3. Первое задание выполнено полностью или частично и для одного из остальных заданий правильно указано значение S . 4. Для второго и третьего задания правильно указаны значения S .	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл	0
<i>Максимальный балл</i>	<i>3</i>

Вариант задания 27 и критерии оценивания
Задание 27. Вариант 1.

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное **чётное** произведение двух показаний, между моментами передачи которых прошло не менее 7 минут. Если получить такое произведение не удаётся, ответ считается равным -1 .

*Вам предлагается два задания, связанных с этой задачей: задание А и задание Б. Вы можете решать оба задания или одно из них по своему выбору. Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание – 0 баллов.*

Задание Б является усложнённым вариантом задания А, оно содержит дополнительные требования к программе.

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ А.

Максимальная оценка за выполнение задания А – 2 балла.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

НАПОМИНАЕМ! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число N – общее количество показаний прибора. Гарантируется, что $N > 7$. В каждой из следующих N строк задаётся одно положительное целое число – очередное показание прибора.

Пример входных данных:

12
12
45
5
3
14
17
23
21
20
19
18
17

Программа должна вывести одно число – описанное в условии произведение, либо -1 , если получить такое произведение не удаётся.

Пример выходных данных для приведённого выше примера входных данных:

54

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Задание Б (решение для задания А приведено ниже, см. программу 4). Чтобы произведение было чётным, хотя бы один сомножитель должен быть чётным, поэтому при поиске подходящих произведений чётные показания прибора можно рассматривать в паре с любыми другими, а нечётные – только с чётными.
--

Для каждого показания с номером k, начиная с $k = 8$, рассмотрим все допустимые по условиям задачи пары, в которых данное показание получено вторым. Минимальное произведение из всех этих пар будет получено, если первым в паре будет взято минимальное подходящее показание среди всех, полученных от начала приёма и до показания с номером $k - 7$. Если очередное показание чётное, минимальное среди предыдущих может быть любым, если нечётное – только чётным.

Для получения эффективного по времени решения нужно по мере ввода данных помнить абсолютное минимальное и минимальное чётное показание на каждый момент времени, каждое вновь полученное показание умножать на соответствующий ему минимум, имевшийся на 7 элементов ранее, и выбрать минимальное из всех таких произведений.
--

Поскольку каждое текущее минимальное показание используется после ввода
--

ещё 7 элементов и после этого становится ненужным, достаточно хранить только 7 последних минимумов. Для этого можно использовать массив из 7 элементов и циклически заполнять его по мере ввода данных. Размер этого массива не зависит от общего количества введённых показаний, поэтому такое решение будет эффективным не только по времени, но и по памяти. Чтобы хранить абсолютный и чётный минимумы, нужно использовать два таких массива.

Ниже приводится пример такой программы, написанной на алгоритмическом языке.

Программа 1. Пример правильной программы на алгоритмическом языке.

Программа эффективна по времени и по памяти

алг

нач

```

цел s = 7           | требуемое расстояние между показаниями
цел аmax = 1001     | больше максимально возможного показания
цел N
ввод N
цел а                | очередное показание прибора
целтаб мини[0:s-1]   | текущие минимумы последних s элементов
целтаб миничет[0:s-1] | чётные минимумы последних s элементов
цел i
| вводим первые s показаний, фиксируем минимумы
цел ма; ма := аmax    | минимальное показание
цел мчет; мчет := аmax | минимальное чётное показание
нц для i от 1 до s
    ввод а
    ма := imin(ма, а)
    если mod(а,2) = 0 то мчет := imin(мчет,а) все
    мини[mod(i, s)] := ма
    миничет[mod(i, s)] := мчет
кц
цел мп = аmax*аmax | минимальное значение произведения
цел п
нц для i от s+1 до N
    ввод а
    если mod(а,2)=0
        то п := а * мини[mod(i, s)]
    иначе если мчет < аmax
        то п := а * миничет[mod(i, s)]
    иначе п := аmax*аmax;
    все
    все
    мп := imin(мп, п)
    ма := imin(ма, а)
    если mod(а,2) = 0 то мчет := imin(мчет,а) все
    мини[mod(i, s)] := ма
    миничет[mod(i, s)] := мчет
кц
если мп = аmax*аmax то мп:=-1 все
вывод мп

```

кон

Возможны и другие реализации. Например, вместо циклического заполнения массива можно каждый раз сдвигать его элементы. В приведённом ниже примере хранятся и сдвигаются не минимумы, а исходные значения. Это требует чуть меньше памяти (достаточно одного массива вместо двух), но по времени решение со сдвигами менее эффективно, чем с циклическим заполнением. Однако время работы остаётся пропорциональным N , поэтому максимальная оценка за такое решение тоже составляет 4 балла.

Программа 2. Пример правильной программы на языке Паскаль.

Программа использует сдвиги, но эффективна по времени и по памяти

```
const s = 7; {требуемое расстояние между показаниями}
      amax = 1001; {больше максимально возможного показания}
var
  N: integer;
  a: array[1..s] of integer; {хранение s показаний прибора}
  a_: integer; {ввод очередного показания}
  ma: integer; {минимальное число без s последних}
  me: integer; {минимальное чётное число без s последних}
  mp: integer; {минимальное значение произведения}
  p: integer;
  i, j: integer;
begin
  readln(N);
  {Ввод первых s чисел}
  for i:=1 to s do readln(a[i]);
  {Ввод остальных значений, поиск минимального произведения}
  ma := amax; me := amax;
  mp := amax*amax;
  for i := s + 1 to N do begin
    readln(a_);
    if a[1] < ma then ma := a[1];
    if (a[1] mod 2 = 0) and (a[1] < me) then me := a[1];
    if a_ mod 2 = 0 then p := a_ * ma
    else if me < amax then p := a_ * me
    else p := amax* amax;
    if (p < mp) then mp := p;
    {сдвигаем элементы вспомогательного массива влево}
    for j := 1 to s - 1 do
      a[j] := a[j + 1];
    a[s] := a_
  end;
  if mp = amax*amax then mp:=-1;
  writeln(mp)
end.
```

Если вместо небольшого массива фиксированного размера (циклического или со сдвигами) хранятся все исходные данные (или все текущие минимумы), программа сохраняет эффективность по времени, но становится неэффективной по памяти, так как требуемая память растёт пропорционально N . Ниже приводится пример такой программы на языке Паскаль. Подобные (и аналогичные по сути) программы оцениваются не выше 3 баллов.

Программа 3. Пример правильной программы на языке Паскаль.

Программа эффективна по времени, но неэффективна по памяти

```
const s = 7; {требуемое расстояние между показаниями}
      amax = 1001; {больше максимально возможного показания}
var
  N, p, i: integer;
  a: array[1..10000] of integer; {все показания прибора}
  ma: integer; {минимальное число без s последних}
  me: integer; {минимальное чётное число без s последних}
  mp: integer; {минимальное значение произведения}
begin
  readln(N);
  {Ввод всех показаний прибора}
  for i:=1 to N do readln(a[i]);
  ma := amax;
  me := amax;
  mp := amax*amax;
  for i := s + 1 to N do
  begin
    if a[i-s] < ma then ma := a[i-s];
    if (a[i-s] mod 2 = 0) and (a[i-s] < me) then
      me := a[i-s];
    if a[i] mod 2 = 0 then p := a[i] * ma
    else if me < amax then p := a[i] * me
    else p := amax * amax;
    if (p < mp) then mp := p
  end;
  if mp = amax*amax then mp := -1;
  writeln(mp)
end.
```

Возможно также переборное решение, в котором находятся произведения всех возможных пар и из них выбирается минимальное. Ниже (см. программу 4) приведён пример подобного решения. Это (и аналогичные ему) решение неэффективно ни по времени, ни по памяти. Оно является решением задания А, но не является решением задания Б. Оценка за такое решение – 2 балла.

Программа 4. Пример правильной программы на языке Паскаль.

Программа неэффективна ни по времени, ни по памяти

```
const s = 7; {требуемое расстояние между показаниями}
var
  N: integer;
  a: array[1..10000] of integer; {все показания прибора}
  mp: integer; {минимальное значение произведения}
  i, j: integer;
begin
  readln(N);
  {Ввод значений прибора}
  for i:=1 to N do
    readln(a[i]);
  mp := 1000 * 1000 + 1;
  for i := 1 to N-s do begin
    for j := i+s to N do begin
      if (a[i]*a[j] mod 2 = 0) and (a[i]*a[j] < mp)
        then mp := a[i]*a[j]
    end;
  end;
  if mp = 1000 * 1000 + 1 then mp := -1;
  writeln(mp)
end.
```

Указания по оцениванию

Баллы

Предварительные замечания.

1. В задаче есть два задания (А и Б). Соответственно, ученик может представить две программы. В каждой из программ должно быть указано, решением какого из заданий она является. Если в работе представлена одна программа, то в ней также должно быть указано, решением какого из заданий она является.

2. Если ученик не указал, к какому заданию относится программа, или можно предположить, что ученик ошибся в идентификации программ, необходимо следовать приведённым ниже инструкциям.

Случай 2.1. Ученик представил только одну программу.

Следует рассматривать программу как решение задания Б и оценивать её по соответствующим критериям.

Случай 2.2. Ученик представил две программы, но указание задания есть только для одной из программ.

Следует рассматривать вторую программу как ответ на оставшееся задание.

Случай 2.3. Ученик представил две программы; ни для одной из них задание не указано, или в обоих решениях указано одно и то же задание.

Следует первую (по порядку в представленных учеником

материалах) программу рассматривать как ответ на задание А, а вторую – как ответ на задание Б. Случай 2.4. Ученик представил более двух программ. Следует рассматривать только две последние программы и соотносить их с заданиями по правилам 2.1–2.3. Случай 2.5. Решение, представленное в качестве решения задания А, по критериям для задания Б может быть оценено в 3 или 4 балла. При этом решение, представленное в качестве решения задания Б, получило меньшую оценку. Следует считать, что ученик перепутал обозначения заданий и оценивать решение, представленное как решение задания А, по критериям задания Б. НАПОМИНАЕМ! <i>Итоговый балл за задачу – это больший из баллов, полученных учеником за каждое из двух представленных решений.</i> <i>Пояснения для проверяющих.</i> 1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то согласно приведённым ниже критериям его оценка будет такой же, как если бы это решение было представлено в качестве решения задания А. 2. Два задания (и, соответственно, возможность для экзаменуемого представить две программы) дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое даёт ему право получить 2 балла, а затем приступить к поиску более эффективного решения. 3. Приведённые в п. 2.1–2.5 правила имеют целью избежать снижения оценки из-за того, что ученик перепутал обозначения заданий	
Критерии оценивания задания А	
Программа решает поставленную задачу для любых соответствующих условию входных данных. Например, допускается переборное решение, аналогичное приведённой выше программе 4. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях оценивания задания Б на 3 балла	2
Не выполнены условия, позволяющие поставить 2 балла. Из описания алгоритма или общей структуры программы видно,	1

что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях	
Не выполнены критерии, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл для задания А</i>	2
Критерии оценивания задания Б	
Программа правильно работает для любых соответствующих условию входных данных и при этом эффективна как по времени, так и по памяти, т.е. не используются массивы и другие структуры данных, размер которых зависит от количества входных элементов, а время работы пропорционально этому количеству. Возможно использование массивов и динамических структур данных (например, контейнеры STL в программе на языке C++) при условии, что в них в каждый момент времени хранится фиксированное количество элементов, требующих для хранения меньше 1кб (минимально необходимое количество – семь; допускается решение с запасом). Программа может содержать не более трёх синтаксических ошибок следующих видов: – пропущен или неверно указан знак пунктуации (запятая, точка с запятой, скобки и т.д.); – неверно написано или пропущено служебное слово языка программирования; – не описана или неверно описана переменная; – применяется операция, недопустимая для соответствующего типа данных. К синтаксическим ошибкам приравнивается использование неверного типа данных (например, использование целого типа вместо вещественного для представления данных при вводе и обработке). Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку	4
Не выполнены условия, позволяющие поставить 4 балла. Программа правильно работает для любых соответствующих условию входных данных, время работы пропорционально количеству входных элементов. Размер используемой памяти не имеет значения и может зависеть от объёма входных данных. В частности, допускается использование одного или нескольких массивов размера N (как в приведённой выше программе 3). Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла. Кроме того, допускается наличие не более одной содержательной ошибки из числа следующих:	3

неверная инициализация при поиске минимального значения; неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при $7 < N < 14$; неточное определение границ массива, выход за границу массива (например, описан массив с границами от 1 до 7, а реально используется от 0 до 6 или наоборот); вычисленный индекс элемента массива на 1 отличается от верного; используется операция "<" вместо "<=", "or" вместо "and" и т.п.; не учитывается, что заданные показания могут начинаться с одного или нескольких чётных чисел; не учитывается, что для данного набора показаний может не быть ни одного удовлетворяющего условиям произведения	
Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Например, допускается переборное решение, аналогичное приведённой выше программе 4. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях на 3 балла	2
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла	0
Максимальный балл для задания Б	4
Итоговый максимальный балл	4

Инструкции и памятки по проверке работ

Задание 24

1. При проверке правильности выполнения пунктов задания, связанных с указанием входных и выходных данных, т.е. ответов на вопросы «Что выведет программа

программа?», «Для какого числа программа работает верно?», «Для какого числа программа работает неверно?», следует учитывать, что в задании не требуется каких-либо обоснований ответов на эти вопросы.

2. При проверке заданий, для которых критериями предусмотрено снижение оценки за указание верной строки в качестве строки с ошибкой нужно подсчитать количество таких строк.

Задание 25

1. Определите, на каком языке записан алгоритм, и при необходимости наведите справки о синтаксисе избранного экзаменуемым языка программирования.
2. Сравните описание алгоритма с имеющимися образцами. В случае совпадения – оцените в соответствии с рекомендациями.
3. Если описание алгоритма не совпадает с образцами, а ошибки в описании алгоритма с первого взгляда не видны, осуществите формальное исполнение алгоритма с тестовыми примерами исходных данных. Длину тестового массива следует сократить до 4–6 элементов. При тестах необходимо особенно тщательно проверять «критические» случаи, например, когда элементы массива одинаковы или изначально упорядочены. Оцените правильность полученных результатов.
4. В случае алгоритма, представленного на русском языке или в виде блок-схемы, необходимо оценить возможность выполнения этого алгоритма человеком и уровень детализации алгоритма (должны выполняться требования дискретности, детерминированности и результативности).
5. При оценке алгоритма отметьте все ошибки, упомянутые в критериях оценивания. В случае, если таких ошибок более двух, сразу снижайте оценку до 0 баллов.
6. Не допускайте произвольного ужесточения критериев оценивания. Не вводите дополнительных ограничений. Не оценивайте синтаксические ошибки, «стиль» программирования, аккуратность записи, наличие комментариев, отступов и прочие важные, но данным заданием не проверяемые вещи.
7. Не забывайте, что эффективность алгоритмов в данной задаче **не оценивается, поэтому не следует снижать оценку за решение, в котором используется два или более просмотра массива.**

Задание 26

1. Сравните последовательно ответы экзаменуемого на вопросы 1а), 1б), 2 и 3 с эталонными.
2. Проверьте наличие и корректность обоснования решений. Стратегия правильного решения в работе экзаменуемого может быть представлена в варианте, отличном от «образцового», но основные положения должны совпадать с предложенными. Графическое оформление дерева игры также может быть различным. Обратите внимание, что для выигрывающего игрока могут быть рассмотрены только выигрышные ходы (стратегия), для проигрывающего – должны быть рассмотрены все варианты его ответов.

Задание 27

1. Определите язык программирования, на котором написана программа. Программы, написанные на языках программирования, отличных от Бейсика и Паскаля тоже должны быть оценены. Это же относится и к современным версиям языков Паскаль (Дельфи) и Бейсик (Вижуал-Бейсик). При необходимости эксперт может воспользоваться справочной литературой.
2. Подсчитайте число синтаксических ошибок в данной программе. Систематически встречающаяся ошибка считается за одну. Так, например, если вместо круглых скобок ученик в записи условий использовал везде квадратные, то это считается за одну ошибку.

3. Рассмотрите реализацию каждой части алгоритма. Определите количество алгоритмических ошибок и количество ошибок в реализации алгоритмов.
4. Если ошибок мало и программа оценивается из 3–4 баллов, то оцените эффективность предложенного решения.

Описание ситуаций сложных для оценивания

Задание 24

Ситуация	Пояснение
Учащийся привел верные примеры входных данных, когда программа работает правильно (неправильно), но не обосновал свой ответ.	Поскольку в задании обоснования не требовалось, за его отсутствие оценка не снижается.
Учащийся привел верные примеры входных данных, когда программа работает правильно (неправильно), обосновал свой ответ, но допустил ошибки в обосновании.	Оценка не снижается.
Доработка программы предложена верно, но в виде фраз на естественном языке, а не на языке программирования.	Если описание содержит чёткие и недвусмысленные указания на необходимые исправления, такое описание следует рассматривать наравне с написанием исправленной строки.

Задание 25

Основная сложность при оценивании этого задания состоит в том, что задание не устанавливает четко, каким образом алгоритм обработки массива может быть записан. Разрешение использовать естественный язык автоматически приводит к употреблению абитуриентами неточных и расплывчатых формулировок, оборотов «аналогично ищем максимальный элемент», «производим подсчет этих чисел и выводим результат» и т. д. При оценке алгоритмов, записанных на естественном языке, основным критерием должна быть возможность их формального исполнения, то есть описание должно быть максимально приближено к записи команд, соответствующих основным операторам языков программирования: должно быть указано, в какой последовательности выполняются инструкции, каково условие завершения цикла и т. д. При этом, например, фраза «Для каждого элемента массива мы выполняем сравнение его с переменной...» может рассматриваться как вполне приемлемое описание цикла с параметром.

При рассмотрении записей алгоритмов на естественном языке следует очень внимательно проследить, как абитуриент описывает операцию присваивания. Наряду с формулой «Присваиваем переменной m значение очередного элемента массива» может употребляться и конструкция «Присваиваем значение очередного элемента массива переменной m » (часто употребляется конструкция «записываем в ...»). Видимо, не стоит снижать оценку за некоторую нечеткость терминологии, если из описания алгоритма видно, что абитуриент ясно представляет себе его выполнение. Если же алгоритм не может быть формально исполнен без каких-то особых разъяснений (формально можно описать это как вызов процедуры или вспомогательного алгоритма), то такой алгоритм должен быть оценен нулем баллов. Пример такого описания: «Сначала находим максимальное значение элементов массива, а затем...». Иногда экзаменуемые пытаются отделаться только общими указаниями, что может быть оценено только нулем баллов.. Например, фраза «Находим максимальные элементы массива и подсчитываем их число» должна быть оценена как недостаточная для формального исполнения в 0 баллов.

Также в 0 баллов следует оценивать применение в программах на языке программирования стандартных функций из библиотек (например, сортировка массива по возрастанию): задание проверяет не знание названий стандартных функций, а знание алгоритмов.

Формулировка задания не предполагает возможности записи алгоритма в виде блок-схемы, но на практике участники иногда рисуют блок-схему алгоритма вместо его описания на формальном или естественном языке. Блок-схемы должны оцениваться по тем же критериям, что и другие формы записи алгоритма. При этом оценка за синтаксические ошибки в блок-схемах (например, запись условия ветвления в блоке-прямоугольнике вместо ромба) не снижается.

Еще одной сложностью при проверке этого задания является разнообразие языков программирования, на которых могут записываться решения: от классического Бейсика с метками и оператором GOTO до C++, PHP и Perl. Это обстоятельство не должно смущать, так как простота формулировки задания предполагает и простую его реализацию на языке программирования, использование небольшого набора базовых операторов, достаточно инвариантных. Задачей экзаменатора при анализе решения **задания** не является проверка знания экзаменуемым синтаксиса языка, а только знания им алгоритмов и умения записать их с определенной степенью формализации.

Таким образом, за синтаксические ошибки в решениях задания оценка не снижается, если только они не ведут к невозможности анализа алгоритма экспертом.

Многие экзаменуемые пишут программы, предусматривающие избыточное количество просмотров массива. За избыточные просмотры оценка не снижается

Задание 26

При проверке решений задания 26 следует учитывать, что правильная стратегия может быть представлена экзаменуемым в самых различных формах (схема, таблица, словесное описание и т.д.). Если решение представлено в виде изображения дерева, то оно может быть схематически изображено способом, отличным от схемы в критериях. Например, корень (исходная позиция) может быть расположен не только слева (как в эталонном ответе), но и сверху и вообще произвольно.

За арифметические ошибки, не приведшие к неверному ответу, оценка не снижается.

Задание 27

1. В решении может присутствовать опечатка, в результате которой оно становится синтаксически или логически неверным ($a[i]$ вместо $a[j]$, $a[i]$ вместо $a[i].x$ и т. п.). Если из анализа решения видно, что это именно опечатка (например, из комментариев, или если в большинстве аналогичных случаев автор программы не ошибся), то ее можно не рассматривать как синтаксическую ошибку.
2. Программа написана на языке программирования, отличном от Бейсика или Паскаля, в например, на языках C#, Java, Perl, PHP и т.д. Такая программа, тем не менее, должна быть оценена. Это же относится и к особенностям современных версий языков Паскаль и Бейсик. При необходимости эксперт может воспользоваться справочной литературой.