

ФЕДЕРАЛЬНАЯ СЛУЖБА ПО НАДЗОРУ В СФЕРЕ ОБРАЗОВАНИЯ И НАУКИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ НАУЧНОЕ УЧРЕЖДЕНИЕ  
«ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»

**Учебно-методические материалы для председателей  
и членов региональных предметных комиссий по проверке  
выполнения заданий с развернутым ответом  
экзаменационных работ ЕГЭ 2014 года**

# **ИНФОРМАТИКА И ИКТ**

## **ЧАСТЬ 3**

**МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ЗАЧЕТА**

Москва  
2014

Учебно-методические материалы для председателей и членов региональных предметных комиссий по проверке выполнения заданий с развернутым ответом экзаменационных работ ЕГЭ 2014 года по информатике и ИКТ подготовлены в соответствии с Тематическим планом работ Федерального государственного научного учреждения «Федеральный институт педагогических измерений», проводимых по заданию Федеральной службы по надзору в сфере образования и науки в 2014 году (в целях научно-методического обеспечения мероприятий общероссийской системы оценки качества образования). 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. Пособие состоит из трех частей.

В первой части («Методические рекомендации по оцениванию выполнения заданий ЕГЭ с развернутым ответом. Информатика и ИКТ») дается краткое описание структуры контрольных измерительных материалов 2014 г. по информатике и ИКТ, характеризуются типы заданий с развернутым ответом, используемые в экзаменационных работах ЕГЭ по информатике и ИКТ и критерии оценки выполнения заданий с развернутым ответом, приводятся примеры оценивания выполнения заданий и даются комментарии, объясняющие выставленную оценку.

Во второй части («Материалы для самостоятельной работы экспертов ЕГЭ по оцениванию выполнения заданий с развернутым ответом. Информатика и ИКТ») в целях организации самостоятельной и групповой работы экспертов дается общий алгоритм работы эксперта, приводятся примеры оценивания выполнения заданий и тексты ответов экзаменуемых.

В третьей части («Материалы для проведения зачета. Информатика и ИКТ») содержатся формулировки заданий с развернутым ответом и приводятся ответы экзаменуемых. Материалы могут быть использованы для самостоятельной работы, а также для проведения зачета.

Авторы будут благодарны за замечания и предложения по совершенствованию пособия.

## Оглавление

<b>УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ПРЕДСЕДАТЕЛЕЙ И ЧЛЕНОВ РЕГИОНАЛЬНЫХ ПРЕДМЕТНЫХ КОМИССИЙ ПО ПРОВЕРКЕ ВЫПОЛНЕНИЯ ЗАДАНИЙ С РАЗВЕРНУТЫМ ОТВЕТОМ ЭКЗАМЕНАЦИОННЫХ РАБОТ ЕГЭ 2014 ГОДА .....</b>	<b>1</b>
<b>ЗАДАНИЯ ДЛЯ ПРОБНОГО ЗАЧЕТА ПО ОЦЕНИВАНИЮ ВЫПОЛНЕНИЯ ЗАДАНИЙ С1 .....</b>	<b>4</b>
ВАРИАНТ 2.....	4
<i>Задание 1. С1. Вариант 2.</i> .....	5
<i>Задание 2. С1. Вариант 2.</i> .....	6
<i>Задание 3. С1. Вариант 2.</i> .....	6
<i>Задание 4. С1. Вариант 2.</i> .....	6
<b>ЗАДАНИЯ ДЛЯ ПРОБНОГО ЗАЧЕТА ПО ОЦЕНИВАНИЮ ВЫПОЛНЕНИЯ С2.....</b>	<b>8</b>
ВАРИАНТ 2.....	8
<i>Задание 1. С2. Вариант 2.</i> .....	10
<i>Задание 2. С2. Вариант 2.</i> .....	10
<i>Задание 3. С2. Вариант 2.</i> .....	11
<i>Задание 4. С2. Вариант 2.</i> .....	11
<b>ЗАДАНИЯ ДЛЯ ПРОБНОГО ЗАЧЕТА ПО ОЦЕНИВАНИЮ ВЫПОЛНЕНИЯ С3.....</b>	<b>11</b>
ВАРИАНТ 2.....	11
<i>Задание 1. С3. Вариант 2.</i> .....	15
<i>Задание 2. С3. Вариант 2.</i> .....	16
<i>Задание 3. С3. Вариант 2.</i> .....	17
<i>Задание 4. С3. Вариант 2.</i> .....	18
<b>ЗАДАНИЯ ДЛЯ ПРОБНОГО ЗАЧЕТА ПО ОЦЕНИВАНИЮ ВЫПОЛНЕНИЯ С4.....</b>	<b>19</b>
ВАРИАНТ 2.....	19
<i>Задание 1. С4. Вариант 2.</i> .....	22
<i>Задание 2. С4. Вариант 2.</i> .....	23
<i>Задание 3. С4. Вариант 2.</i> .....	24
<i>Задание 4. С4. Вариант 2.</i> .....	25
<b>ЗАДАНИЯ ДЛЯ ИТОГОВОГО ЗАЧЕТА ПО ОЦЕНИВАНИЮ ЭКЗАМЕНАЦИОННЫХ РАБОТ.....</b>	<b>26</b>
ВАРИАНТ 3.....	26
<i>Работа 1. Вариант 3.</i> .....	37
<i>Работа 2. Вариант 3.</i> .....	40
<i>Работа 3. Вариант 3.</i> .....	45
<i>Работа 4. Вариант 3.</i> .....	48
<i>Работа 5. Вариант 3.</i> .....	51
ВАРИАНТ 4.....	54
<i>Работа 1. Вариант 4.</i> .....	64
<i>Работа 2. Вариант 4.</i> .....	67
<i>Работа 3. Вариант 4.</i> .....	69
<i>Работа 5. Вариант 4.</i> .....	75

## Задания для пробного зачета по оцениванию выполнения заданий С1

### 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 &gt; 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 &gt; 1 do         begin             N := N div 10;             sum := sum + 1;         end;     writeln(sum); end.         </pre>
Си	Алгоритмический язык
<pre> #include&lt;stdio.h&gt; int main() {     long int N;     int sum;     scanf("%ld", &amp;N);     sum = 1;     while (N &gt; 1)     {         N = N /10;         sum = sum + 1;     }     printf("%d", sum); }         </pre>	<pre> алг нач     цел N, sum     ввод N     sum := 1     нц пока N &gt; 1         N := div(N, 10)         sum := sum + 1     кц     вывод 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)

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

### Задание 1. С1. Вариант 2.

Оцените решение:

C 1.

- 1) 4
- 2) 1
- 3) а)  $sum := 0;$   
б)  $while\ N > 0\ do$

Задание 2. C1. Вариант 2.

Оцените решение:

C1)  
① 938 при этом  $sum = 4$ .  
②  $= 2$   
в)  $while\ N \geq 1; sum := 0$ .

Задание 3. C1. Вариант 2.

Оцените решение:

C1 ① Три вводе числа 938, программа выдает: 4.  
② Для числа 134 программа будет работать верно и выдает: 3.  
③  $var\ N: longint;$   
1.  $sum: integer;$   
2.  $begin$   
3.  $readln\ (N);$   
4.  $sum := 1;$   
5.  $while\ N > 1\ do$   
6.  $begin$   
7.  $N := N \div 10;$   
8.  $sum := sum + 1;$   
9.  $end;$   
10.  $writeln\ (sum);$   
11.  $end.$   
Ошибки:  
5.  $sum := 0;$   
6.  $while\ N >= 1\ do$

Задание 4. C1. Вариант 2.

Оцените решение:

C<sub>1</sub>.

1) 4

2) 0

3) sum := 1 ;      sum := 0 ;  
while N > 1 do      while N > 0 do

## Задания для пробного зачета по оцениванию выполнения С2

### 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 &lt;stdio.h&gt; #define N 30 void main(void) { int a[N]; inti, x, y; float s; for (i=0; i&lt;N; i++) scanf("%d", &amp;a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 30 <u>целтаба</u> [1:N] <u>цел</u> i, x, y <u>вещ</u> s <u>нц</u> для i от 1 до N   <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Русский (естественный) язык	
<p>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. ...</p>	

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



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).

<b>Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)</b>	
<b>На языке Паскаль</b>	<b>На языке Бейсик</b>
<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>
<b>На языке СИ</b>	<b>На Алгоритмическом языке</b>
<pre>x=0; y=0; for (i=0; i&lt;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 для i от 1 до N</u> <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>выводs</u></pre>
<b>На русском (естественном) языке</b>	
<p>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.</p> <p>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.</p> <p>После цикла производим деление счетчика суммы X на счетчик количества Y и записываем результат в переменную S. Выводим значение переменной S.</p>	

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 нечетность. 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
Максимальный балл	2

### Задание 1. C2. Вариант 2.

Оцените решение:

```

X:=0;
Y:=0;
S:=0;
for i:=1 to N do
    if (a[i] mod 2=1) then
        X:=X+a[i];
Y:=Y+1;
S:=X/N;
writeln(S);

```

### Задание 2. C2. Вариант 2.

Оцените решение:

```

x:=0;

```

```

y:=0;
float x1,y1;
for (i=0; i<N; i++)
if (a[i]%2==1)
{
    x=x+a[i];
    y++;
}
x1 = (float)x;
y1 = (float)y;
s=x/y;
printf( s);

```

### Задание 3. С2. Вариант 2.

#### Оцените решение:

```

x,y= 0
for i:=1 to N do
if ((a[i] +1) mod 2)=0 then begin
    x:=x+a[i];
    inc(y);
end;
s:=x/y;
print(s);

```

### Задание 4. С2. Вариант 2.

#### Оцените решение:

```

X = 0
Y = 0
FOR I = 1 TO N
    S = A(I) / 2
    IF (INT(S) – S)*(INT(S) – S)>0 THEN
        X = X + A(I)
        Y = Y + 1
    ENDIF
NEXT I
S = X / Y
PRINT S

```

### Задания для пробного зачета по оцениванию выполнения С3

#### 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.

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.

<p align="center"><b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>	
<p>(допускаются иные формулировки ответа, не искажающие его смысла)</p> <ol style="list-style-type: none"> <li>1. а) Петя может выиграть первым ходом, если <math>S=10, \dots, 29</math>. Во всех случаях нужно утроить количество камней в куче. При меньших значениях <math>S</math> 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 <math>S=9</math> 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.</li> <li>2. Возможные значения <math>S</math>: 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.</li> <li>3. Возможное значение <math>S</math>: 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).</li> </ol>	

	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)
<b>7</b>	$7+1=8$	$8+1=9$	$9+1=10$	$10*3=30$
			$9*3=27$	$27*3=81$
	$7*3=21$	$21*3=63$		

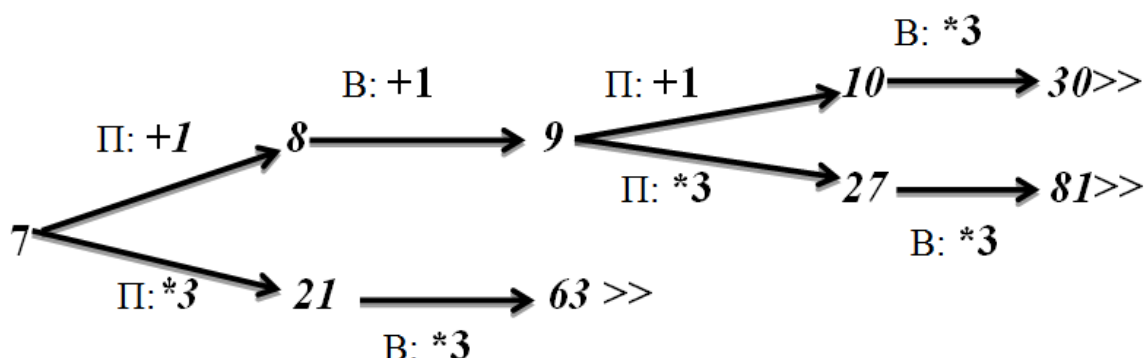


Рис.1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>Ошибка в решении, не искажающая основного замысла, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.</p> <p>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.</p> <p>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.</p> <p>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</p>	

<p>возможных партий.</p> <p>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.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения или другим способом.</p>	
<p>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).</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> <li>1. Задание 3 выполнено полностью.</li> <li>2. Первое и второе задания выполнены полностью.</li> <li>3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения <math>S</math>.</li> </ol>	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> <li>1. Первое задание выполнено полностью.</li> <li>2. Во втором задании правильно указано одно из двух возможных значений <math>S</math>, и для этого значения указана и обоснована выигрышная стратегия Пети.</li> <li>3. Первое задание выполнено полностью или частично и для одного из остальных заданий правильно указано значение <math>S</math>.</li> <li>4. Для второго и третьего задания правильно указаны значения <math>S</math>.</li> </ol>	1
<p>Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл</p>	0
<p><i>Максимальный балл</i></p>	3

Задание 1. С3. Вариант 2.

Оцените решение:

(С3)

1) а) от 90 до 29  
 б)  $S' = 8 \rightarrow 9$

2)  $S = 8$

3)  $N = 7$

Стратегия Банки:

Исх.	Пени.	Возна	Платя	Возна
7	+1 8	+1 = 9	10 или 27	30 или 81 <del>счит</del>
7	$\times 3$ 21	$\times 3 = 63$		

Два случая

Получен.

Задание 2. С3. Вариант 2.

Оцените решение:

C3

1) ~~Утро~~  
Утром nadie meno больше

2) 9, Петя выигрывает одним ходом.

	П	В выигрывает
9	10	30
	27	81

$S = 9$

2. 8, а также 3  
У ~~Петя~~ <sup>Петя</sup> будет завершающий  
победный ход

3. 7. Победный ход будет у Васи.



Задание 3. С3. Вариант 2.

Оцените решение:

C3

1 а) Петя может выиграть  
первым ходом при  $S=10$  ( $\times 3$ )  
и при  $S=29$  ( $+1$ )

б) Вова может выиграть при  
остальных ходах Пети,  
например,  $S=9$ .

2. Петя может выиграть,  
если Вова пойдет макс, то  
будет от 10 до 29 копеек

	10	28	29
Вова	$\times 3$	$\downarrow$	$+1$
	30	84	30
	★	★	★

Задание 4. С3. Вариант 2.

Оцените решение:

C3 | 1A

$S = 10 \dots 29$  тк. можно  
только в этих случаях выигрывает  
1-й ходим, упрощенно.

15 При  $S=9$  Вова выигрывает своим  
первым ходом, независимо от  
хода Пети

2. Петя выигрывает при старте  
из позиции 3, 8, тк. из  
них следует 9.

## Задания для пробного зачета по оцениванию выполнения С4

### Вариант 2.

На вход программе подаются 366 строк, которые содержат информацию о среднесуточной температуре всех дней 2004 года. Формат каждой из строк следующий: сначала записана дата в виде dd.mm (на запись номера дня и номера месяца в числовом формате отводится строго два символа, день от месяца отделен точкой), затем через пробел записано значение температуры — число со знаком плюс или минус, с точностью до 1 цифры после десятичной точки. Данная информация отсортирована по значению температуры, то есть хронологический порядок нарушен. Требуется написать программу на языке Паскаль или Бейсик (или любом другом языке программирования), которая будет выводить на экран информацию о месяце (месяцах) среднемесячная температура у которого (которых) наименее отклоняется от среднегодовой. В первой строке вывести среднегодовую температуру. Найденные значения для каждого из месяцев следует выводить в отдельной строке в виде: номер месяца, значение среднемесячной температуры, отклонение от среднегодовой температуры.

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
Программа читает входные данные, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев, одновременно суммируя все температуры в году. Затем с использованием этого массива ищется минимальное отклонение среднемесячной температуры от среднегодовой. За дополнительный просмотр этого массива распечатывается информация об искомым месяцах. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, месяц с минимальным отклонением единственен).	
<b>Указания по оцениванию</b>	<b>Баллы</b>
Программа работает верно, т.е. определяет все месяцы, в которых среднемесячная температура минимально отклоняется от среднегодовой, не содержит вложенных циклов, в тексте программы не анализируется каждый месяц в отдельности. Допускается наличие в тексте программы одной синтаксической ошибки. <b>Пример правильной и эффективной программы на языке Паскаль:</b> <pre>Const d:array[1..12] of integer =     (31,29,31,30,31,30,31,31,30,31,30,31); var tm:array[1..12] of real;     m:1..12;     data:string[5];     min,ty,t:real;     i:integer; begin     for i:=1 to 12 do tm[i]:=0;     ty:=0;{среднегодовая температура}     for i:=1 to 366 do         begin             readln(data,t); {вычисляемномермесяца} m:=(ord(data[4])-ord('0'))*10             +ord(data[5])-ord('0');             tm[m]:=tm[m]+t;             ty:=ty+t;</pre>	4

```

end;
for i:=1 to 12 do
    tm[i]:=tm[i]/d[i];
ty:=ty/366;
min:=100;
for i:=1 to 12 do
    if abs(tm[i]-ty)<min then
        min:=abs(tm[i]-ty);
writeln('Среднегодовая температура = ',ty:0:2);
for i:=1 to 12 do
    if abs(abs(tm[i]-ty)-min)<0.0001 then
        writeln(i, ' ',tm[i]:0:2, ' ',abs(tm[i]-ty):0:2);
readln
end.

```

**Пример правильной и эффективной программы на языке Бейсик:**

```

DATA 31,29,31,30,31,30,31,31,30,31,30,31
DIM i, m, d(12) AS INTEGER
DIM tm(12)
DIM dat AS STRING *5
FOR i = 1 TO 12
    tm(i) = 0
    READ d(i)
NEXT i
ty = 0
'ty - среднегодовая температура
FOR i = 1 TO 366
    INPUT dat, t
    'Вычисляем номер месяца
    m=(ASC(MID$(dat,4,1))-ASC("0"))*10+
    ASC(MID$(dat,5,1))-ASC("0")
    tm(m) = tm(m) + t
    ty = ty + t
NEXT i
FOR i = 1 TO 12
    tm(i) = tm(i) / d(i)
NEXT i
ty = ty / 366
min = 100
FOR i = 1 TO 12
    IF ABS(tm(i) - ty) < min THEN min = ABS(tm(i) - ty)
NEXT i
PRINT "Среднегодовая температура = ";
PRINT USING "##.##"; ty
FOR i = 1 TO 12
    IF ABS(ABS(tm(i) - ty) - min) < .0001 THEN
        PRINT i;
        PRINT USING "##.##    "; tm(i); ABS(tm(i) - ty)
    END IF
NEXT i
END

```

Программа составлена верно, но содержит вложенные циклы (от 1 до 12 и от 1 до 366), возможно, запоминает значения всех температур в массиве. Допускается наличие от одной до трех синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная	3
Программа составлена в целом верно с вложенными циклами или без них, или обрабатывает каждый месяц явным образом (12 операторов IF или оператор CASE, содержащий 12 вариантов), или предварительно сортирует входные данные в хронологическом порядке. Возможно, выводит значение только одного месяца с минимальным отклонением температуры. Возможно в реализации алгоритма содержатся 1–2 ошибки (используется знак “<” вместо “>”, “or” вместо “and” и т.п.). Допускается наличие не более пяти синтаксических ошибок.	2
Программа, возможно, некорректно определяет номер месяца или неверно вводит или выводит данные, или содержит ошибку в алгоритме поиска минимума или средней температуры, или отклонение берется не по абсолютной величине. Допускается наличие не более семи синтаксических ошибок.	1
Задание не выполнено или выполнено неверно	0
<i>Максимальный балл</i>	4

Задание 1. С4. Вариант 2.

Оцените решение:

```
Program S4roki; const n:=360
var a: array [1..12] of integer;
    b: array [1..12] of integer;
    c: real;
    m, z, i, t, k, min, SrG: integer;

begin
    z:=0;

    for i:=1 to n do
        begin
            read(c);
            m:=(round(c*100)) mod 100;
            read(t);
            for k:=1 to 12 do
                if m=k then
                    begin inc(b[k]);
                        a[k]:=a[k]+t;
                    end;
                z:=z+t;
            end;
            min:=32000;
            SrG:=(z*100) div n;
            for i:=1 to 12 do
                begin
                    a[i]:=(a[i]*100) div (b[i]);
                    if abs(SrG-a[i])<min then min:=a[i];
                end;
                write(SrG/100:0:1);
            for i:=1 to 12 do
                if abs(SrG-a[i])=min then
                    writeln(i,'L', a[i]/100:0:1, 'L',
                        'L', abs(SrG-a[i])/100:0:1);
            end;
        end;
```

## Задание 2. С4. Вариант 2.

Оцените решение:

```
+ypz.  
month = record  
    temp, days: integer;  
end;  
  
var  
    a: array[1..12] of month;  
    g: string[3]; m: string[2];  
    i, temp, srg, min: integer;  
  
begin  
    for i := 1 to 366 do begin  
        readln(g, m, temp);  
        a[strtointint(m)].temp := a[strtointint(m)].temp + temp;  
        a[strtointint(m)].days := a[strtointint(m)].days + 1;  
        srg := srg + temp;  
    end;  
  
    srg := srg / 366;  
  
    for i := 1 to 12 do  
        begin  
            a[i].temp := srg - a[i].temp / a[i].days;  
            if abs(a[i]) <= min then min := abs(a[i]);  
        end;  
  
        for  
        writeln(srg);  
        for i := 1 to 12 do  
            begin  
            if abs(a[i]) = min then  
                writeln(i, ' ', a[i] + srg, a[i]);  
  
        end.
```

### Задание 3. С4. Вариант 2.

Оцените решение:

```

const eps = 1e-8;

var t, sumt: real;
    m: array[1..12] of real;
    numday: array[1..12] of integer;
    ch: char;
    i, mini, mon: integer;
Begin
    sumt := 0;
    fillchar(m, sizeof(m), 0); fillchar(numday, sizeof(numday), 0);
    for i := 1 to 366 do begin
        read(ch); read(ch); read(ch);
        read(mon);
        read(t);
        sumt := sumt + t;
        m[mon] := m[mon] + t;
        inc(numday[mon]);
        readln;
    end;
    sumt := sumt / 366;
    for i := 1 to 12 do
        m[i] := m[i] / numday[i];
    mini := 1;
    for i := 2 to 12 do
        if abs(m[i] - sumt) < abs(m[mini] - sumt) then mini := i;
    writeln(sumt: 0: 1);
    for i := 1 to 12 do
        if abs(abs(m[i] - sumt) - abs(m[mini] - sumt)) < eps then
            writeln(i, ' ', m[i]: 0: 1, ' ', abs(m[i] - sumt));
    end;

```



Задание 4. С4. Вариант 2.

Оцените решение:

```
Var
e: char;
dm: array[1..31, 1..12] of integer;
numofday: array[1..12] of integer;
sum: array[1..12] of integer;
i, j, m, t, all: integer;
A: extended;

Begin
  for i := 1 to 12 do numofday[i] := 0;
  for i := 1 to 366 do
    begin
      readln(d, e, m, e, t);
      dm[d, m] := t;
      numofday[m] := numofday[m] + 1;
    end;
  for i := 1 to 12 begin
    begin
      for j := 1 to 31 do sum[i] := dm[j, i];
      all := sum[i];
    end;
    writeln(A := all / 366;
    writeln(e:0:1);
    for i := 1 to 12 do
      begin
        e := sum(i) / numofday[i];
        writeln(e:0:1);
      end;
  end.
end.
```

## Задания для итогового зачета по оцениванию экзаменационных работ

### Вариант 3.

С1

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 = 9 WHILE N &gt;= 10     digit = N MOD 10     IF digit &lt; 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 := 9;     while N &gt;= 10 do         begin             digit := N mod 10;             if digit &lt; min_digit then                 min_digit := digit;             N := N div 10;         end;         writeln(digit);     end.         </pre>
Си	Алгоритмический язык
<pre> #include&lt;stdio.h&gt; int main() {     long int N;     int digit, min_digit;     scanf("%ld", &amp;N);     min_digit = 9;     while (N &gt;= 10)     {         digit = N % 10;         if (digit &lt; 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 &gt;= 10         digit := mod(N, 10)         если digit &lt; 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. 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.

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.</p> <p>1. Программа выведет число 4.</p> <p>2. Первая ошибка. Неверное условие окончания цикла. Программа не будет рассматривать старшую цифру числа.</p> <p>Строка с ошибкой:  <code>while N &gt;= 10 do</code></p> <p>Возможные варианты исправления:  <code>while (N &gt;= 1)</code>                      или  <code>while (N &gt; 0)</code></p> <p>При этом замены на  <code>while (N &gt; 1)</code> или <code>while (N &gt;= 0)</code>                      корректными не являются</p> <p>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.</p> <p>Строка с ошибкой:  <code>writeln(digit);</code></p> <p>Необходимо в строке с выводом результата заменить <code>digit</code> на <code>min_digit</code></p>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
<p>Обратите внимание! В задаче требовалось выполнить <b>три</b> действия: указать, что выведет программа при конкретном входном значении, и исправить две ошибки.</p> <p>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия (ниже указано, какое действие считается выполненным).</p> <p>1. Верно указано, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указано на неверное условие окончания цикла, и оно исправлено на верное.</p> <p>3. Указана и верно исправлена ошибка вывода переменной <code>digit</code> вместо <code>min_digit</code></p>	

Правильно выполнены все пункты задания. Программа после исправлений для всех натуральных чисел $N$ , не превосходящих $10^9$ , верно определяет минимальную цифру. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения	3
1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в п. 1 задания ответ не приведён или приведён неверно, или выполнен п. 1, и верно исправлена только одна ошибка). Верное указание на ошибку при её неверном исправлении при этом не засчитывается. 2. Или выполнен п. 1, а вместо указания на ошибки в программе и их исправления приведён новый верный текст решения, возможно, совершенно непохожий на исходный. 3. Или правильно выполнены все действия (приведён верный ответ на вопрос 1, и исправлены обе ошибки), но в текст программы внесены и другие изменения, приводящие к её неверной работе	2
Правильно выполнено только одно действие из трёх, то есть либо только выполнен п. 1, либо он не выполнен или выполнен неверно и верно исправлена только одна ошибка программы путём её явного указания и исправления	1
Все пункты задания выполнены неверно (ответ на п. 1 не приведён или приведён неверно; ошибки не найдены или найдены, но не исправлены или исправлены неверно)	0
<i>Максимальный балл</i>	<i>3</i>

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от –1000 до 1000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди положительных элементов массива, имеющих чётное значение. Если в исходном массиве нет элемента, значение которого чётно и положительно, то вывести сообщение «Не найдено».

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Паскаль
<pre> N = 20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I  ...  END </pre>	<pre> const     N = 20; var     a: array [1..N] of integer;     i, j, min: integer; begin     for i := 1 to N do         readln(a[i]);     ... end. </pre>
Си	Алгоритмический язык
<pre> #include &lt;stdio.h&gt; #define N 20 void main() {     int a[N];     int i, j, min;     for (i = 0; i &lt; N; i++)         scanf("%d", &amp;a[i]);     ... } </pre>	<pre> алг нач     цел N = 20     целтаб a[1:N]     цел i, j, min     нц для i от 1 до N         ввод a[i]     кц     ... кон </pre>
Естественный язык	
<p>Объявляем массив A из 20 элементов.</p> <p>Объявляем целочисленные переменные I, J, MIN.</p> <p>В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й.</p> <p>...</p>	

В качестве ответа Вам 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, Free Pascal 2.4) 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	
<pre> min := 1001; for i := 1 to N do     if (a[i]&gt;0) and (a[i] mod 2=0) and (a[i]&lt;min) then         min := a[i]; if min &lt; 1001 then writeln(min) else writeln('Не найдено');</pre>	
На алгоритмическом языке	
<pre> min := 1001 нц для i от 1 до N     если a[i]&gt;0 и mod(a[i],2)=0 и a[i]&lt;min         то             min := a[i]     все кц если min &lt; 1001     то         вывод min иначе     вывод "Не найдено" все</pre>	
На языке Бейсик	
<pre> MIN = 1001 FOR I = 1 TO N     IF A(I)&gt;0 AND A(I) MOD 2=0 AND A(I)&lt;MIN THEN         MIN = A(I)     END IF NEXT I IF MIN &lt; 1001 THEN     PRINT MIN ELSE     PRINT "Не найдено" END IF</pre>	

На языке Си	
<pre> min = 1001; for (i = 0; i&lt;N; i++)     if (a[i]&gt;0 &amp;&amp; a[i]%2==0 &amp;&amp; a[i]&lt;min)         min = a[i]; if (min&lt;1001)     printf("%d", min); else     printf("Не найдено"); </pre>	
На естественном языке	
<p>Записываем в переменную MIN начальное значение, равное 1001. В цикле от первого элемента до двадцатого находим остаток от деления элемента исходного массива на 2. Если значение данного остатка равно 0 и значение текущего элемента массива больше 0, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.</p> <p>После завершения цикла проверяем значение переменной MIN. Если оно меньше 1001, то выводим его, иначе выводим сообщение «Не найдено»</p>	
Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение.</p> <p>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</p>	2
<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1) не инициализируется или неверно инициализируется переменная MIN (например, присваивается начальное значение, меньшее или равное 1000);</li> <li>2) при выводе ответа не учитывается, что требуемого числа в массиве может не быть;</li> <li>3) отсутствует вывод ответа в случае существования минимального числа, удовлетворяющего условию задачи;</li> <li>4) в сравнении с 0 вместо знака «больше» используется знак «больше или равно»;</li> <li>5) неверно осуществляется проверка делимости на 2;</li> <li>6) на делимость на 2 проверяется не значение элемента, а его индекс;</li> <li>7) в сложном условии вместо логической операции «И» используется логическая операция «ИЛИ»;</li> <li>8) используется переменная, не объявленная в разделе описания переменных;</li> <li>9) не указано или неверно указано условие завершения цикла;</li> <li>10) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;</li> <li>11) неверно расставлены операторные скобки</li> </ol>	1
Ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно	0
Максимальный балл	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 **один** камень или увеличить количество камней в куче в **три** 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 39. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 39 или больше камней.

В начальный момент в куче было  $S$  камней,  $1 \leq S \leq 38$ .

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.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа  $S$ , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрывающий ход для каждого указанного значения  $S$ .  
 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 таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.
3. Укажите значение  $S$ , при котором:
  - у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и
  - у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.
 Для указанного значения  $S$  опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах – количество камней в куче.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
1. а) Петя может выиграть, если $S = 13, \dots, 38$ . Во всех этих случаях достаточно утроить количество камней. При меньших значениях $S$ за один ход нельзя получить кучу, в которой больше 38 камней.	
б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 12$ камней. Тогда после первого хода Пети в куче будет 13 или 36 камней. В обоих случаях Ваня утраивает количество камней и выигрывает в один ход.	
2. Возможные значения $S$ : 4 и 11. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 12 камней: в первом случае утроением, во втором добавлением одного камня. 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.	
3. Возможное значение $S$ : 10. После первого хода Пети в куче будет 11 или 30 камней. Если в куче станет 30 камней, Ваня утроит количество камней и выиграет первым ходом. Ситуация, когда в куче 11 камней, уже 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)
10	$10+1=11$	$11+1=12$	$12+1=13$	$13*3=39$
			$12*3=36$	$36*3=108$
	$10*3=30$	$30*3=90$		

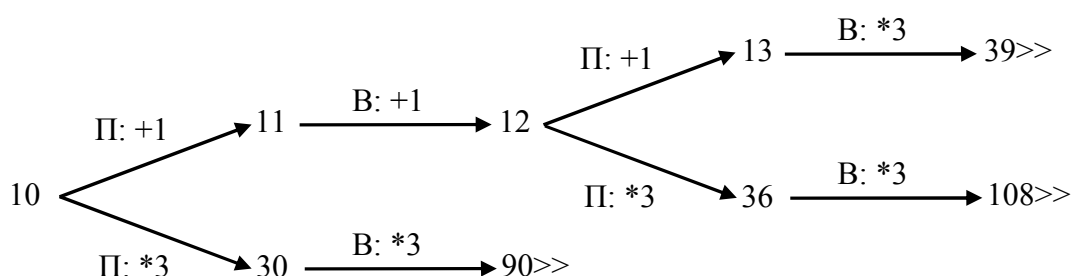


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три задания. Их трудность возрастает. 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>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 полностью, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, то есть показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если:</p> <ul style="list-style-type: none"> <li>а) правильно указаны все позиции, в которых Петя выигрывает первым ходом;</li> <li>б) правильно указана позиция, в которой Ваня выигрывает первым ходом, и явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от выполненного полностью задания состоит в том, что не указаны явно ходы, которыми выигрывает Петя или Ваня.</li> </ul> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех партий, возможных при выбранной стратегии</p>	

<p>Пети.</p> <p>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.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения или другим способом</p>	
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
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> <li>1. Третье задание выполнено полностью.</li> <li>2. Первое и второе задания выполнены полностью.</li> <li>3. Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения <math>S</math></li> </ol>	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> <li>1. Первое или второе задание выполнено полностью.</li> <li>2. Во втором задании правильно указано одно из двух возможных значений <math>S</math>, и для этого значения указана и обоснована выигрышная стратегия Пети.</li> <li>3. Первое задание выполнено частично, и для одного из остальных заданий правильно указаны значения <math>S</math>.</li> <li>4. Для второго и третьего заданий правильно указаны значения <math>S</math></li> </ol>	1
Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл	0
Максимальный балл	3

#### C4

На ускорителе для большого числа частиц производятся замеры скорости каждой из них. Скорость частицы – это целое неотрицательное число. Частиц, скорость которых измерена, может быть очень много, но не может быть меньше трёх. Скорости всех частиц различны.

При обработке результатов в каждой серии эксперимента отбирается основное множество скоростей. Это непустое подмножество скоростей частиц (в него могут войти как скорость одной частицы, так и скорости всех частиц серии), такое, что сумма значений скоростей у него чётна и максимальна среди всех возможных непустых подмножеств с чётной суммой. Если таких подмножеств несколько, то из них выбирается то подмножество, которое содержит наименьшее количество элементов.

Вам предлагается написать эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет обрабатывать результаты эксперимента, находя основное множество.

Перед текстом программы кратко опишите используемый Вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество частиц  $N$ . В каждой из последующих  $N$  строк записано одно целое неотрицательное число, не превышающее  $10^9$ . Все  $N$  чисел различны.

*Пример входных данных:*

```
5
123
2
```

1000  
0  
10

Программа должна вывести в порядке возрастания номера частиц, скорости которых принадлежат основному множеству данной серии. Нумерация частиц ведётся с единицы.

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

2 3 5

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Основное множество состоит из всех значений скоростей, кроме 0, если он встречается, и кроме минимального нечётного значения, если таких значений нечётное число.</p> <p>Программа читает все входные данные один раз, не запоминая все входные данные в массиве, размер которого равен <math>N</math>. Во время чтения данных запоминается номер 0, если он встретится (по условию все значения различны, поэтому 0 встречается не больше одного раза), подсчитывается количество нечётных значений и ищется минимальное нечётное значение. После окончания ввода распечатываются все номера, кроме номера 0 и номера минимального нечётного значения, но только в случае, если их количество нечётно.</p> <p>Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования</p>	
<b>Пример правильной и эффективной программы на языке Паскаль:</b>	
<pre>var n,i,j,k,c,min,a: longint; begin   readln(n);   min := 1000000001;   k := 0;   j := 0;   c := 0;   for i := 1 to n do     begin       readln(a);       if a = 0 then j := i;       if a mod 2 &lt;&gt; 0 then         begin           c := c + 1;           if a &lt; min then             begin               min := a;               k := i;             end           end         end       end;   for i := 1 to n do     if (i &lt;&gt; j) and ((c mod 2 = 0) or (i &lt;&gt; k)) then       write(i, ' ');   end.</pre>	

**Пример правильной и эффективной программы на языке Бейсик:**

```
INPUT n
min = 0
```

<pre> k = 0 j = 0 c = 0 FOR i = 1 TO n   INPUT a   IF a = 0 THEN j = i   IF a MOD 2 &lt;&gt; 0 THEN     c = c + 1     IF (min = 0) OR (a &lt; min) THEN       min = a       k = i     END IF   END IF NEXT i FOR i = 1 TO n   IF (i &lt;&gt; j) AND ((c MOD 2 = 0) OR (i &lt;&gt; k)) THEN PRINT i NEXT i END </pre>	
Указания по оцениванию	Баллы
<p>Программа верно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу <math>N</math> (числу частиц). Программа просматривает входные данные один раз, определяя номер 0, количество нечётных значений и минимальное нечётное число. Затем распечатываются все номера частиц, кроме частицы с нулевым значением, а в случае, когда количество нечётных значений нечётно, и кроме номера частицы с минимальным нечётным значением. Допускается наличие в тексте программы до трёх синтаксических ошибок следующих видов: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку)</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве или другой структуре данных (например, контейнер <code>priority_queue</code>, <code>vector</code>, <code>set</code> или <code>map</code> в C++), размер которого соответствует числу <math>N</math>. Этот массив, или массив отобранных номеров, возможно, потом сортируется). При этом общая сложность алгоритма не превышает <math>CN^2</math>, где <math>C</math> – константа, не зависящая от <math>N</math>. Допускается наличие до пяти синтаксических ошибок, описанных выше.</p> <p>Кроме того, допускается наличие одной содержательной ошибки, например ошибки из следующего списка:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (при условии, что в целом ввод организован правильно);</li> <li>2) программа неправильно работает при больших значениях введённых чисел (наступает переполнение);</li> <li>3) допущена ошибка в реализации алгоритма сортировки;</li> <li>4) используется “&lt;” вместо “&lt;=”, “AND” вместо “OR” и т.п.</li> </ol>	3
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Например, программа использует алгоритм перебора всех возможных подмножеств и сравнивает суммы значений элементов подмножеств. Допускается до семи синтаксических ошибок и не более двух содержательных ошибок (см.</p>	2

примеры 3 балла)	в	критериях	на	
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом выполнено одно из двух условий. 1. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. 2. Программа правильно работает в одном из важных частных случаев. Допускается любое количество синтаксических ошибок				1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла				0
		Максимальный балл		4

### Работа 1. Вариант 3.

Оцените выполнение заданий C1-C4:

C1.	
1. Программа выведет число "4".	
2. 1) $\alpha$ ) while $\sum \neq 10$ do	
$\delta$ ) writeln (digit);	
2) $\alpha$ ) while $\sum \geq 1$ do	
$\delta$ ) writeln (min-digit);	

## C2 / qBasic /

```

N = 20
DIM A(N) AS INTEGER
DIM I, j, MIN AS INTEGER
FOR I = 1 TO N
    INPUT A(I)
NEXT I

```

$j = 3$  // в переменную  $j$  вводим заведомо неверное число.

for  $i = 1$  to  $N$  // открываем цикл от 1 до  $N$

~~min = A(i)~~ // Переменной  $min$  присваиваем значения  
этой массива

if ( ~~$A(i) > 0$~~ ) // проверяем на положительность.

and ( $A(i) \bmod 2 = 0$ ) // проверяем на четность

and ( $A(i) < j$  or  $j = 3$ ) // проверяем на минимальность, но  
если оно не минимальное, то  
т.е.  $j = 3$  прокруч. цикл и одознач., это  
т.е. еще не найден.

then  $j = A(i)$  // если всё выполн. присваиваем перемен.  
 $j$  значение эл.  $A(i)$

next (i) // закрываем цикл.

if  $j <= 3$  // число исконое найдено и не равно заведомо  
ложному

then PRINT  $j$  // выводим миним. на печать.

else PRINT "не найдено"

Так, запишу программу, которую было необходимо записать  
без лишних пояснений и нарушения синтаксиса:

```

j = 3
FOR i = 1 TO N
    IF (A(i) > 0) AND (A(i) MOD 2 = 0) AND (A(i) < j OR j = 3) THEN
        j = A(i)
    NEXT i
    IF j <= 3 THEN PRINT j
    ELSE PRINT "не найдено"

```

C3.

- а) от 13 до 38 (увеличить кол-во камней в 3 раза и добавить в эту сумму камней при 38)  
от 12 (Теме добавляет один камень → Вова увеличивает в 3 раза. Тема увеличивает в 3 раза  
Вова увеличивает в 3 раза)
- а) 2, 4  
б) 4

C4.

```

Program C4;
Const b=10
Var
  st, ab, s: String;
in, r, n, i, j, k: integer;
  naz: array [1.. b] of record
    nm: String;
    kol: integer;
  end;
  x: array [1.. 4] of integer;
begin
  readln(ab); readln(n); {Because}
  for i:=1 to n do begin
    for j:=1 to 4 do
      x[j]:=0;
    read(s);
    p:=0;
    for j:=1 to length(s) do
      if (copy(s, j, 1) = ' ')
        then p:=p+1;
  end;

```

```

  if (p+1) = length(ab)
  then BEGIN
    k:=0;
    for j:=1 to length(s) do
      if (copy(s, j, 1) = ' ')
        then begin
          k:=k+1;
          x[k]:=j;
        end;
    j:=1;
    while (j<=10) do begin
      for z:=1 to p do
        if (copy(s, 1, 1) = copy(ab, 1, 1)) AND ~
          ~AND (copy(s, x[z]+1, 1) = copy(ab, z+1, 1))
        then r:=1
        else r:=0;
      if (r=1)
      then
        if (naz[j].nm='')
        then begin naz[j].nm:=s;
          naz[j].kol:=naz[j].kol+1
        end
      end;

```

Ex. (Borland Pascal 7.0)

```
Var N, j, i, k, c, min, a: longint;  
begin  
  readln(N);  
  min := 1000000001;  
  k := 0;  
  c := 0;  
  j := 0;  
  for i := 1 to N do begin  
    readln(a);  
    if (a = 0) then j := i;  
    if (a mod 2 <> 0) then begin  
      min := a;  
      k := i;  
    end; end; end;  
  for i := 1 to N do  
    if (i < j) and ((c mod 2 = 0) or (i < k)) then write(i, ' ');  
  end.
```

---

Работа 2. Вариант 3.

Оцените выполнение заданий C1-C4



С1. 1. При вводе числа 547 программа выведет 4, что является правильным

2. Ошибка заключается в строке:

while  $N \geq 10$  do

Так как программа не читает <sup>(старшую)</sup> последнюю цифру  
и например при вводе числа 123 выведет 2

Исправить её можно так:

while  $N > 0$  do

Теперь программа будет работать правильно

С2.

end;

if  $a[1] > 0$  then

$j := a[1] \bmod 2$ ;

if  $j = 0$  then

$min := a[1]$ ;

for  $i = 1$  to  $N$  do

$j := a[i] \bmod 2$

if  $(a[i] > 0)$  and  $(j = 0)$  and  $(a[i] < min)$  then

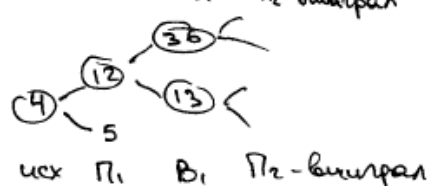
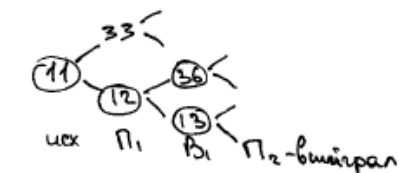
$min := a[i]$ ;

end;

writeln(min);

- С3. 1) При  $S \geq 13$  Петя выигрывает в один ход. В любом случае он увеличит кучу в 3 раза и выиграет.  
 2) При  $13 \leq S \leq 38$   
 3) При  $S = 12$  Петя Вам выигрывает у Пети в любом случае если сделает ход  $\cdot 3$  иначе при  $\cdot 1$  он проигрывает.

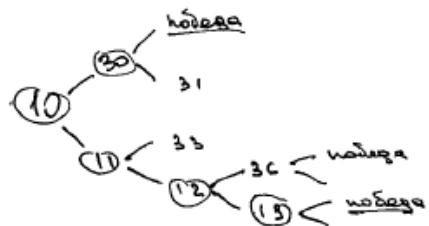
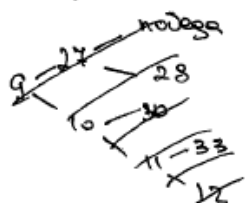
2)  $S = 11$  и 4



При  $S = 11$  Петя должен сделать ход  $\cdot 1$  тогда при любом ходе Вами Петя выигрывает 2м ходом.

При  $S = 4$  Петя должен сделать ход  $\cdot 3$  тогда при любом ходе Вами Петя выигрывает 2м ходом.

3) При  $S = 10$



Исх  $P_1$   $V_1$   $P_2$   $V_2$

При недостатке места для ответа используйте обратную сторону бланка

Вам может выиграть при  $S = 10$  первым ходом умножив на 3  $\cdot 3$  если Петя сделает 1й ход  $\cdot 3$ .  
 Или же Вам выигрывает 2 ходом.  
 При условии что Петя первым ходом делает  $\cdot 1$ , затем Вам первым ходом делается  $\cdot 1$ , тогда в данной ситуации При любом 2-м ходе Пети Вам 2м ходом ~~уже~~ делается ход  $\cdot 3$ .

```

C4. import java.util.*;
import java.io.*;
public class Task {
    public static void main (String[] args) throws
    IOException {
        BufferedReader reader = new BufferedReader
        (InputStreamReader (System.in));
        int N = Integer.parseInt (reader.readLine());
        int a[] = new Integer [N];
        for (int i=0; i<N; i++) {
            int a[i] = Integer.parseInt (reader.readLine());
        }
    }
}

```

Java 1.6.0\_27  
javac.exe Task.java

См. на opposite →

```

int minn = 0; int k;
for (int j = 0; j < N; j++) {
if (j == 0) { minn = 0
    if (minn == 0 and a[j] > 0) { minn = a[j]; }
    if (a[j] > 0 and a[j]/2 == 1 and minn > a[j]) {
        minn = a[j];
        k = j;
    }
}

int S = 0; int l = 0;
int n[] = new Integer[N];
for (int p = 0; p < N; p++) {
    if (a[p] > 0) {
        S = S + a[p];
        n[p] = p + 1;
        l = l + 1;
    }
}

if (S/2 == 0) { for (int z = 0; z < l; z++) {
    System.out.p System.out.println(n[z]); }
} else { for (int x = 0; x < l; x++) {
    if (x != k) { System.out.println(n[x]); }
}
}
}

```

and -  
 - on globalni ammuksini  
 number "1"

Работа 3. Вариант 3.

Оцените выполнение заданий C1-C4:

C1. 1. Вывод программы: 4

2. 1) Ошибка в программе:

$N := N \text{ div } 10;$

2) Правильный вариант строки:

$N := N \bmod 10;$

C2. 

```
min = 1001;
for j := 1 to n do
begin
  if (a[j] > 0) and (a[j] mod 2 = 0) and (a[j] < min) then min := a[j];
end;
if min = 1001 then writeln('Не найдено') else writeln(min);
```

C3.1.а) Петя может выиграть в один ход если в куче будет нечетно 13 и более камней. ( $S \geq 13$ ), у него-либо количество втрое;

б) Ваня может выиграть своим первым ходом при любом ходе Пети, если в куче будет нечетно 12 камней ( $S = 12$ );

2. Петя может выиграть в один или двумя своим вторым ходом, не зависимо от хода Вани, при  $S = 4$  и  $S = 11$ ;

3. При  $S = 10$  Ваня имеет выигрышную стратегию, при которой он может победить как первым, так и вторым своим ходом.

$S$	П	→	В	→	П	→	В	→
10	$\cdot 3$	30	$\cdot 3$	(90)				
	$+1$	11	$+1$	12	$+1$	13	$\cdot 3$	(39)
					$\cdot 3$	36	$\cdot 3$	(108)



```

    if  $x \bmod 2 = 1$  then begin
        write(k);
        if  $x < \text{min}$  then begin
             $\text{min} := x$ ;
             $\text{num} := i$ ;
        end;
    end;
end;
if  $x = 0$  then begin
     $z := i$ ;  $a[i] := \text{true}$ ;
end;
 $\text{sum} := \text{sum} + x$ ;
end;
if  $k \bmod 2 = 1$  then begin
     $\text{sum} := \text{sum} - \text{min}$ ;
     $a[\text{num}] := \text{true}$ ;
end;
 $q := \text{false}$ ;
for  $i := 1$  to  $n$  do
    if not  $a[i]$  then begin
        write(i);
         $q := \text{true}$ ;
    end;
if not  $q$  then
    write(z);
End.

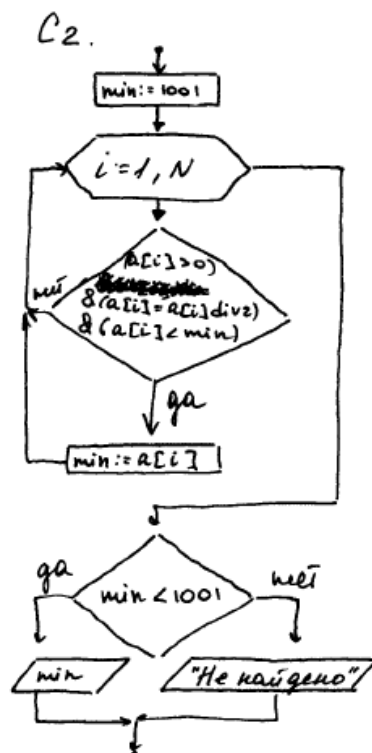
```

---

# Работа 4. Вариант 3.

Оцените выполнение заданий C1-C4:

- C1. 1) При вводе числа 547 выводится число 4  
 2) (Паскаль) I. writeln(digit); - ошибка, т.к. должно выводиться минимальное значение, а не текущее.  
 Правильная запись: writeln(min-digit);  
 II, while N >= 10 do - ошибка, т.к. при такой записи не учитываются натур. числа из промежутка [1; 9].  
 Правильная запись: while N >= 1 do



Free Pascal

```

min := 1001
for i := 1 to N do begin
  if (a[i] > 0) AND (a[i] = a[i] div 2)
    AND (a[i] < min) then min := a[i]
end;
if min < 1001 then write(min)
else write('не найден')
...
  
```

C3

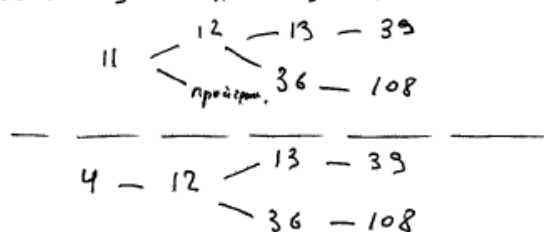
1. а) Петя может выиграть в один ход, если начальное кол-во камней в куче будет в промежутке от 13 до 38 штук, при таком раскладе Петя должен увеличить кол-во камней в трое, и тогда он победит.

б. Ваня выигрывает своим вторым ходом, если изначально кол-во камней будет ~~14~~ 12. (S=12)

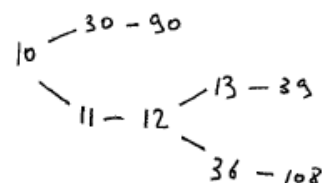
$$\begin{array}{r}
 11 \mid 13 \times 3 \rightarrow 39 \\
 12 \mid 13 \times 3 \rightarrow 39 \\
 12 \mid 36 \times 3 \rightarrow 108
 \end{array}$$



2. Петя выиграет своим вторым ходом, если первоначальное кол-во камней  $S$  будет либо 4, либо 11.

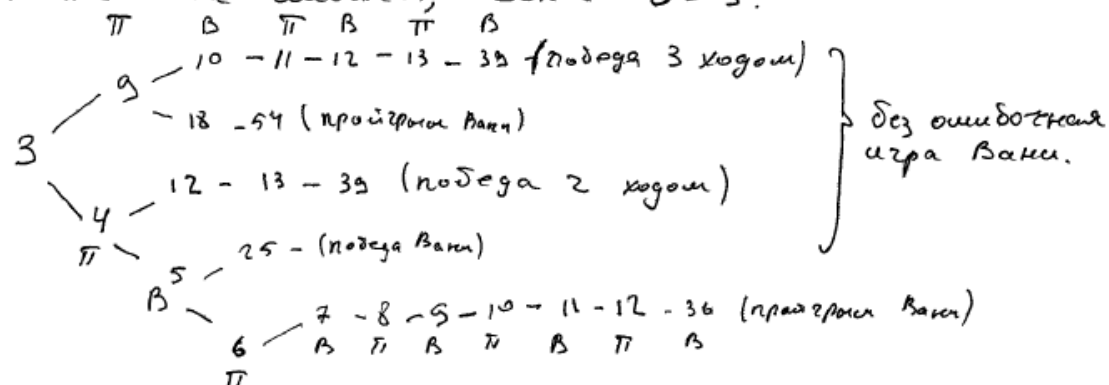


3. а. У Вани есть выигрышная стратегия, позволяющая выигрывать 1 или 2 ходами, при  $S=10$ .



- б. Гарантированно выиграть первым ходом

Ваня не сможет, если  $S=3$ .



Ответ: 1. а)  $[13; 38]$

б) 12

2. 4, 11

3. а) 10

б) 3

C4: #include <stdio.h>

```
#struct list t
```

```
long int v;
```

```
struct list *next; }
```

```
struct list *Head;
```

```
void main ()
```

```
{ int N, i; long int x;
```

```
struct list *f, *t;
```

```
scanf("%d", &N);
```

```
Head = malloc malloc(sizeof(struct list)); // выделение памяти для
```

```
Head->next = NULL;
```

```
scanf("%ld", &x);
```

```
Head->v = x;
```

```
f = Head;
```

```
for (i=0; i<N; i++)
```

```
{ t = malloc(sizeof(struct list));
```

```
f->next = t;
```

```
scanf scanf("%ld", &x);
```

```
t->v = x;
```

```
t->next = NULL;
```

```
f = t; }
```

```
f = Head;
```

Алгоритм: считываем N, далее  
через список (связанный) заносим  
ем входными скоростями. далее  
смотрим, какие из них четные  
и выводим их номера.

```
for (i=1; i<=N; i++)
```

```
{ if (f->v % 2 == 0)
```

```
printf("%d", i);
```

```
f = f->next;
```

```
}
```

```
}
```

### Работа 5. Вариант 3.

Оцените выполнение заданий C1-C3:

C1. 1) 4

- 2) а) while  $N \geq 10$  do - неправильно  
while  $N \geq 0$  do - правильно  
б) writeln(digit); - неправильно  
writeln(min - digit); - правильно.

C2

```
min := 1001;
```

```
for i := 1 to 20 do
```

```
  if (a[i] mod 2 = 0) and (a[i] > 0) and (a[i] < min)
```

```
    then min := a[i];
```

```
if min = 1001 then write("Не найдено")
```

```
  else write(min);
```

(присваиваем переменной отвечающей за значение минимального числа 1001, так как больше 1000 значений быть не может. после нахождения min, проверим равно ли 1001, если да, то нужных значений в массиве нет.)

C3 1.  $39:3=13$

a)  $S \geq 13$  т.к.  $13 \cdot 3 = 39$ ;  $14 \cdot 3 = 42$ ;  $38 \cdot 3 = 114$ .

для  $S \geq 13$  выигрышный ход  $\times 3$ ; для  $S=38$  выигрышный ход  $\times 3 + 1$

б) Петя не выигрывает первым ходом при  $S \leq 13$ , но Ваня выигрывает.

при  $1 \leq S \leq 13$   $S = 12$

При любом ходе Пети ход Вани:  $\times 3$

~~$4+1=5$~~ ,  ~~$5 \times 3$~~   $12+1=13$ ;  $13 \times 3=39$ .

$12 \times 3=36$ ;  $36 \times 3=108$

2.  $S = 11$

~~$S = 10$~~  ~~Петя:  $10+1=11$~~  ~~Ваня:  $11 \times 3=33$~~

~~Петя:  $10+1=11$~~

~~Ваня:  $11 \times 3=33$~~

~~Петя:  $33 \times 3=99$~~

$S=11$  Петя:  $11+1=12$

Ваня:  $12 \times 3=36$  или  $12+1=13$

Петя:  $36 \times 3=108$  или  $13 \times 3=39$ .

3.  $S=10$

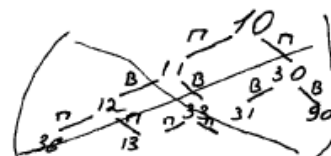
П:  $10+1=11$  или  $10 \times 3=30$

В:  $11+1=12$  или  $30 \times 3=90$ .

П:  $12+1=13$

В:  $13 \times 3=39$ .

~~$S=11$  Петя:  $11+1=12$  или  $11 \times 3=33$~~   
~~Ваня:  $12 \times 3=36$~~   ~~$33 \times 3=99$~~

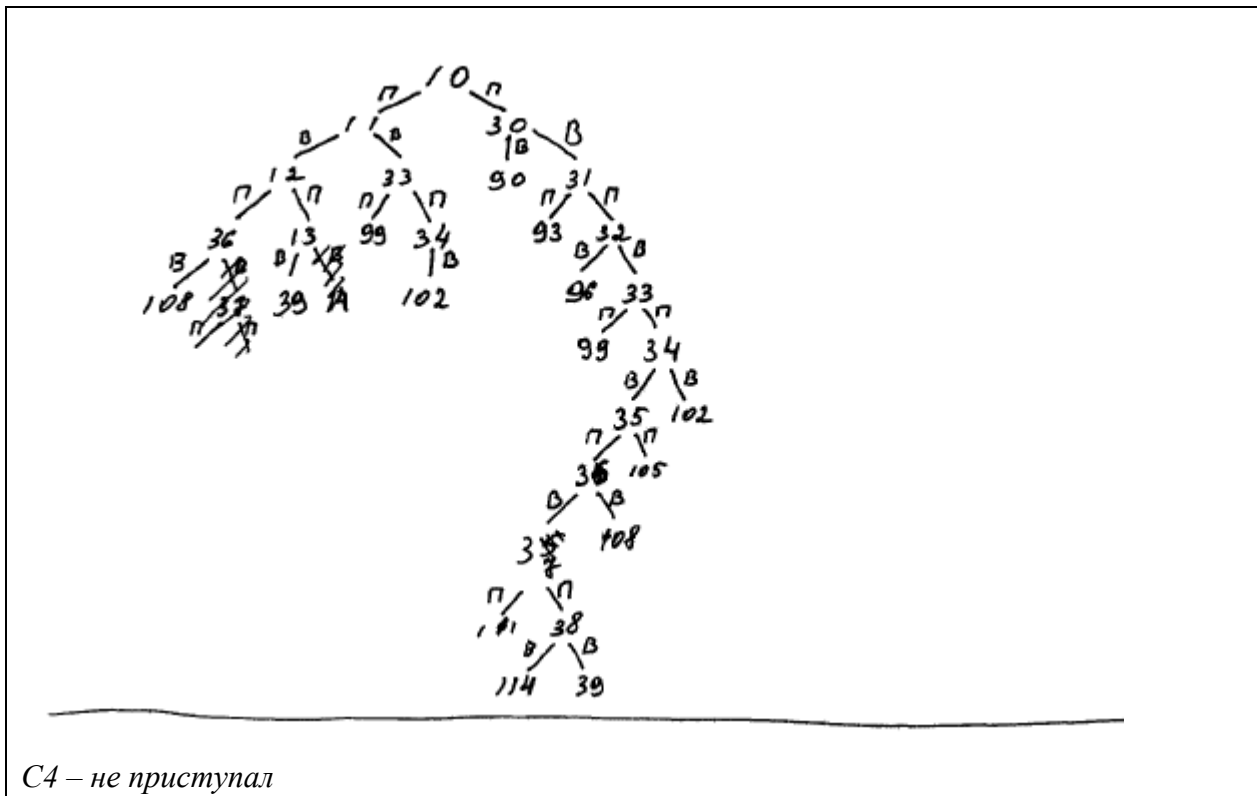


Ответ: 1. a)  $S \geq 13$ ; б)  $S=12$

2.  $S=11$

3.  $S=10$

см. на дод.



C4 – не приступал

## Вариант 4.

C1

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число  $N$ , не превосходящее  $10^9$ , и выводится максимальная 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 max_digit = 10 WHILE N &gt; 9     digit = N MOD 10     IF digit &gt; max_digit THEN         max_digit = digit     END IF     N = N \ 10 WEND PRINT max_digit END         </pre>	<pre> var N: longint;     digit, max_digit: integer; begin     readln(N);     max_digit := 10;     while N &gt; 9 do         begin             digit := N mod 10;             if digit &gt; max_digit then                 max_digit := digit;             N := N div 10;         end;     writeln(max_digit); end.         </pre>
Си	Алгоритмический язык
<pre> #include&lt;stdio.h&gt; int main() {     long int N;     int digit, max_digit;     scanf("%ld", &amp;N);     max_digit = 10;     while (N &gt; 9)     {         digit = N % 10;         if (digit &gt; max_digit)             max_digit = digit;         N = N / 10;     }     printf("%d", max_digit); }         </pre>	<pre> алг нач     цел N, digit, max_digit     ввод N     max_digit := 10     нц пока N &gt; 9         digit := mod(N, 10)         если digit &gt; max_digit то             max_digit := digit         все         N := div(N, 10)     кц     вывод max_digit кон         </pre>

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 542.
2. 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.

<p align="center"><b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>	
<p>Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.</p> <p>1. Программа выведет число 10.</p> <p>2. Первая ошибка. Неверная инициализация ответа (переменная <code>max digit</code>).</p> <p>Строка с ошибкой:</p> <pre>max_digit := 10;</pre> <p>Возможные варианты исправления:</p> <pre>max_digit := 0;</pre> <p>Возможны и другие исправления инициализации, например на отрицательное число, в том числе <code>-maxint</code>.</p> <p>3. Вторая ошибка. Неверное условие продолжения цикла. Программа не будет рассматривать старшую цифру числа.</p> <p>Строка с ошибкой:</p> <pre>while N &gt; 9 do</pre> <p>Возможные варианты исправления:</p> <pre>while (N &gt;= 1) do</pre> <p>или</p> <pre>while (N &gt; 0) do</pre> <p>При этом замены на</p> <pre>while (N &gt; 1) do</pre> <p>или</p> <pre>while (N &gt;= 0) do</pre> <p>корректными не являются</p>	
Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>три</b> действия: указать, что выведет программа при конкретном входном значении, и исправить две ошибки.</p> <p>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия (ниже указано, какое действие считается выполненным).</p> <p>1. Верно указано, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указана и верно исправлена ошибка инициализации (не обязательно с упоминанием этого термина).</p> <p>3. Указано на неверное условие продолжения цикла, и оно исправлено на верное.</p> <p>Каждый из п. 2 и 3 считается выполненным, если:</p> <ul style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа</li> </ul>	
<p>Правильно выполнены все пункты задания. Программа после исправлений для всех натуральных чисел <math>N</math>, не превосходящих <math>10^9</math>, верно определяет максимальную цифру.</p> <p>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения</p>	3
<p>1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в п. 1 задания ответ неверный или отсутствует, или выполнен п. 1, и верно исправлена только одна ошибка). Верное указание на ошибку при её</p>	2

<p>неверном исправлении при этом не засчитывается.</p> <p>2. Или выполнен п. 1, а вместо указания на ошибки в программе и их исправления приведён новый верный текст решения, возможно, совершенно непохожий на исходный.</p> <p>3. Или правильно выполнены все действия (приведён верный ответ на вопрос 1, и исправлены обе ошибки), но в текст программы внесены и другие изменения, приводящие к её неверной работе</p>	
Правильно выполнено только одно действие из трёх, то есть либо только выполнен п. 1, либо он не выполнен или выполнен неверно и верно исправлена только одна ошибка программы путём её явного указания и исправления	1
Все пункты задания выполнены неверно (ответ на п. 1 не приведён или приведён неверно; ошибки не найдены или найдены, но не исправлены или исправлены неверно)	0
<i>Максимальный балл</i>	<i>3</i>

## C2

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от  $-1000$  до  $1000$  включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди положительных элементов массива, кратных 5. Если в исходном массиве нет элемента, значение которого положительно и делится на 5, то вывести сообщение «Не найдено».

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

<b>Бейсик</b>	<b>Паскаль</b>
<pre> N = 20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N     INPUT A(I) NEXT I ... END </pre>	<pre> const     N = 20; var     a: array [1..N] of integer;     i, j, min: integer; begin     for i := 1 to N do         readln(a[i]);     ... end. </pre>
<b>Си</b>	<b>Алгоритмический язык</b>
<pre> #include &lt;stdio.h&gt; #define N 20 void main() {     int a[N];     int i, j, min;     for (i = 0; i &lt; N; i++)         scanf("%d", &amp;a[i]);     ... } </pre>	<pre> алг нач     цел N = 20     целтаб a[1:N]     цел i, j, min     нц для i от 1 до N         ввод a[i]     кц     ... кон </pre>
<b>Естественный язык</b>	
Объявляем массив A из 20 элементов.	



Объявляем целочисленные переменные I, J, MIN.  
 В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й.  
 ...

В качестве ответа Вам 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, Free Pascal 2.4) 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	
<pre>min := 1001; for i := 1 to N do     if (a[i]&gt;0) and (a[i] mod 5=0) and (a[i]&lt;min) then         min := a[i]; if min &lt; 1001 then writeln(min) else writeln('Не найдено');</pre>	
На алгоритмическом языке	
<pre>min := 1001 нц для i от 1 до N     если a[i]&gt;0 и mod(a[i],5)=0 и a[i]&lt;min         то             min := a[i]     все кц если min &lt; 1001     то         вывод min иначе     вывод "Не найдено" все</pre>	
На языке Бейсик	
<pre>MIN = 1001 FOR I = 1 TO N     IF A(I)&gt;0 AND A(I) MOD 5=0 AND A(I)&lt;MIN THEN         MIN = A(I)     END IF NEXT I IF MIN &lt; 1001 THEN     PRINT MIN ELSE     PRINT "Не найдено" END IF</pre>	

На языке Си	
<pre> min = 1001; for (i = 0; i&lt;N; i++)     if (a[i]&gt;0 &amp;&amp; a[i]%5==0 &amp;&amp; a[i]&lt;min)         min = a[i]; if (min&lt;1001)     printf("%d", min); else     printf("Не найдено"); </pre>	
На естественном языке	
<p>Записываем в переменную MIN начальное значение, равное 1001. В цикле от первого элемента до двадцатого находим остаток от деления элемента исходного массива на 5. Если значение данного остатка равно 0 и значение текущего элемента массива больше 0, то сравниваем значение текущего элемента массива со значением переменной MIN. Если текущий элемент массива меньше MIN, то записываем в MIN значение этого элемента массива. Переходим к следующему элементу.</p> <p>После завершения цикла проверяем значение переменной MIN. Если оно меньше 1001, то выводим его, иначе выводим сообщение «Не найдено»</p>	
Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение.</p> <p>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</p>	2
<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1) не инициализируется или неверно инициализируется переменная MIN (например, присваивается начальное значение, меньшее или равное 1000);</li> <li>2) при выводе ответа не учитывается, что требуемого числа в массиве может не быть;</li> <li>3) отсутствует вывод ответа в случае существования минимального числа, удовлетворяющего условию задачи;</li> <li>4) в сравнении с 0 вместо знака «больше» используется знак «больше или равно»;</li> <li>5) неверно осуществляется проверка делимости на 5;</li> <li>6) на делимость на 5 проверяется не значение элемента, а его индекс;</li> <li>7) в сложном условии вместо логической операции «И» используется логическая операция «ИЛИ»;</li> <li>8) используется переменная, не объявленная в разделе описания переменных;</li> <li>9) не указано или неверно указано условие завершения цикла;</li> <li>10) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;</li> <li>11) неверно расставлены операторные скобки</li> </ol>	1
Ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно	0
Максимальный балл	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 **один** камень или увеличить количество камней в куче в **три** 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 48. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 48 или больше камней.

В начальный момент в куче было  $S$  камней,  $1 \leq S \leq 47$ .

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.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа  $S$ , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения  $S$ , и укажите выигрышающий ход для каждого указанного значения  $S$ .  
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 таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.

3. Укажите значение  $S$ , при котором:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

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

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
1. а) Петя может выиграть, если $S = 16, \dots, 47$ . Во всех этих случаях достаточно утроить количество камней. При меньших значениях $S$ за один ход нельзя получить кучу, в которой больше 47 камней.	
б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 15$ камней. Тогда после первого хода Пети в куче будет 16 или 45 камней. В обоих случаях Ваня утраивает количество камней и выигрывает в один ход.	
2. Возможные значения $S$ : 5 и 14. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 15 камней: в первом случае утроением, во втором добавлением одного камня. 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.	
3. Возможное значение $S$ : 13. После первого хода Пети в куче будет 14 или 39 камней. Если в куче станет 39 камней, Ваня утроит количество камней и выигрывает первым ходом. Ситуация, когда в куче 14 камней, уже разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим	

вторым ходом.

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

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)
13	$13+1=14$	$14+1=15$	$15+1=16$	<u><math>16*3=48</math></u>
			$15*3=45$	<u><math>45*3=135</math></u>
	$13*3=39$	<u><math>39*3=117</math></u>		

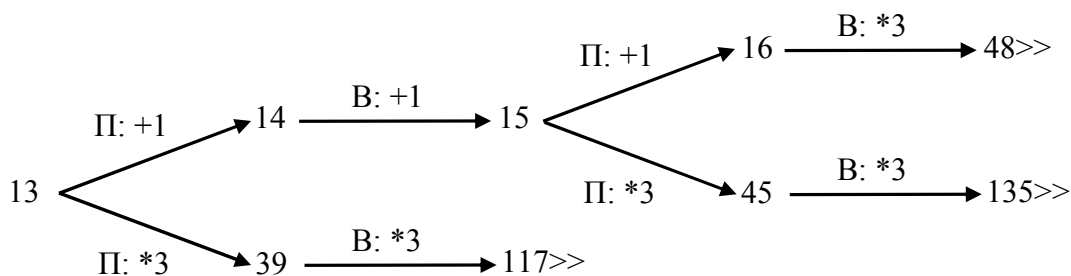


Рис. 1. Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

Указания по оцениванию	Баллы
<p>В задаче от ученика требуется выполнить три задания. Их трудность возрастает. 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>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 полностью, если правильно указана позиция, в которой Ваня выигрывает первым ходом, и описана стратегия Вани, то есть показано, как Ваня может получить кучу, в которой содержится нужное количество камней, при любом ходе Пети.</p> <p>Первое задание считается выполненным частично, если: а) правильно указаны все позиции, в которых Петя выигрывает первым ходом; б) правильно указана позиция, в которой Ваня выигрывает первым ходом, и явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от выполненного полностью задания состоит в том, что не указаны явно ходы, которыми выиграет Петя или Ваня.</p> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех партий, возможных при выбранной стратегии Пети.</p> <p>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</p>	

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. Третье задание выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения $S$	2
Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий. 1. Первое или второе 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 частично, и для одного из остальных заданий правильно указаны значения $S$ . 4. Для второго и третьего заданий 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>

#### C4

На ускорителе для большого числа частиц производятся замеры скорости каждой из них. Скорость частицы – это целое число (положительное, отрицательное или 0). Частиц, скорость которых измерена, может быть очень много, но не может быть меньше трёх. Скорости всех частиц различны. В серии обязательно присутствует хотя бы одна частица с отрицательной скоростью. При обработке результатов в каждой серии эксперимента отбирается основное множество скоростей. Это такое непустое подмножество скоростей частиц (в него могут войти как скорость одной частицы, так и скорости всех частиц серии), для которого произведение скоростей является минимальным среди всех возможных подмножеств. При нахождении произведения знак числа учитывается. Если есть несколько таких множеств, то берётся то, которое содержит наибольшее количество элементов.

Вам предлагается написать эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет обрабатывать результаты эксперимента, находя основное множество.

Перед текстом программы кратко опишите используемый Вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество частиц  $N$ . В каждой из последующих  $N$  строк записано одно целое число, по абсолютной величине не превышающее  $10^9$ . Все  $N$  чисел различны.

*Пример входных данных:*

```
5
123
2
-1000
0
10
```

Программа должна вывести в порядке возрастания номера частиц, скорости которых принадлежат основному множеству данной серии. Нумерация частиц ведётся с единицы.

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

1 2 3 5

<p align="center"><b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)</p>
<p>Основное множество состоит из всех значений скоростей, кроме 0, если он встречается, и кроме минимальной по модулю отрицательной скорости, если отрицательных скоростей чётное число.</p> <p>Программа читает все входные данные один раз, не запоминая все входные данные в массиве, размер которого равен <math>N</math>. Во время чтения данных запоминается номер 0, если он встретится (по условию все значения различны, поэтому 0 встречается не больше одного раза), подсчитывается количество отрицательных значений и ищется минимальное по модулю отрицательное значение. После окончания ввода распечатываются все номера, кроме номера 0 и номера минимального по модулю отрицательного значения, но только в случае, если их чётное число.</p> <p>Ниже приведены примеры решения задания на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования</p>
<p><b>Пример правильной и эффективной программы на языке Паскаль:</b></p>
<pre>var n, i, j, k, c, min, a: longint; begin   readln(n);   min := -1000000001;   k := 0;   j := 0;   c := 0;   for i := 1 to n do     begin       readln(a);       if a = 0 then j := i;       if a &lt; 0 then         begin           c := c + 1;           if a &gt; min then             begin               min := a;               k := i;             end           end         end;       end;   for i := 1 to n do     if (i &lt;&gt; j) and ((c mod 2 &lt;&gt; 0) or (i &lt;&gt; k)) then       write(i, ' ');   end.</pre>

Пример правильной и эффективной программы на языке Бейсик:		
<pre> INPUT n min = 0 k = 0 j = 0 c = 0 FOR i = 1 TO n   INPUT a   IF a = 0 THEN j = i   IF a &lt; 0 THEN     c = c + 1     IF (min = 0) OR (a &gt; min) THEN       min = a       k = i     END IF   END IF NEXT i FOR i = 1 TO n   IF (i &lt;&gt; j) AND ((c MOD 2 &lt;&gt; 0) OR (i &lt;&gt; k)) THEN PRINT i NEXT i END </pre>		
Указания по оцениванию		Баллы
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу <math>N</math> (числу частиц). Программа просматривает входные данные один раз, определяя номер 0, количество отрицательных значений и минимальное по модулю отрицательное число. Затем распечатываются все номера частиц, кроме частицы с нулевым значением, а в случае, когда отрицательных значений чётное число, и кроме номера частицы с минимальным по модулю отрицательным значением. Допускается наличие в тексте программы до трёх синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку)</p>		4
<p>Программа работает верно, но входные данные запоминаются в массиве или другой структуре данных (например, контейнер <code>priority_queue</code>, <code>vector</code>, <code>set</code> или <code>map</code> в C++), размер которого растёт с ростом количества частиц. Этот массив, или массив отобранных номеров, возможно, потом сортируется). При этом общая сложность алгоритма не превышает <math>CN^2</math>, где <math>C</math> – константа, не зависящая от <math>N</math>. Допускается наличие до пяти синтаксических ошибок, описанных выше.</p> <p>Кроме того, допускается наличие одной содержательной ошибки, например ошибки из следующего списка:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (при условии, что в целом ввод организован правильно);</li> <li>2) программа неправильно работает при больших значениях введённых чисел (наступает переполнение);</li> <li>3) допущена ошибка в реализации алгоритма сортировки;</li> <li>4) используется “&lt;” вместо “&lt;=”, “AND” вместо “OR” и т.п.</li> </ol>		3

Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Например, программа использует алгоритм перебора всех возможных подмножеств и сравнивает произведения значений элементов подмножеств. Допускается до семи синтаксических ошибок и не более двух содержательных ошибок (см. примеры в критериях на 3 балла)	2
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом выполнено одно из двух условий. 1. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи. 2. Программа правильно работает в одном из важных частных случаев. Допускается любое количество синтаксических ошибок	1
Не выполнены условия, позволяющие поставить 1, 2, 3 или 4 балла	0
Максимальный балл	4

#### Работа 1. Вариант 4.

Оцените выполнение заданий C1-C4:

<p>C1. 1) begin  digit := 542 mod 10;  if digit &gt; max_digit then  max_digit := digit;  N := 542 div 10;  end;  write ln (max_digit);  end.  N = 54    max_digit = 10</p> <p>2) if digit &gt; max digit  это не может быть, т.к. строка digit := 542 mod 10 означает, что digit равно или меньше остатку от деления на 10, не может быть больше 9.  Значит, надо просто изменить знак неравенства.  if digit &lt; max digit.</p> <p>C2.  if (a[i] &gt;= 1000) and (a[i] &lt;= -1000) then  write ('ошибка')  else  min := 1000;  if a[i] mod 5 = 0 then begin  j := a[i];  if min &gt; j then  min := j;  end;  end.</p>	<p>digit := 54 mod 10;  if digit &gt; max_digit then  max_digit := digit;  N := 54 div 10;  end;  write ln (max_digit);  end.  N = 5    max_digit = 10.</p>
--	---



С3 1) а) Выбранными 5 для Пети будут являться числа от 16 до 47 (включительно), любое из этих чисел умноженное на 3 будет больше 48. В случае с числом 47 достаточно прибавить 1.

б) Если же Пете увеличим число камней на 1, то выбранное 9 для Вани будут являться числом от 19 до 46.

Если Теме увеличим число камней в 3 раза, то выбранное 5 для Вани являться числом от 6 до 15.

В обоих случаях Ваня пользуется стратегией увеличения камней в 3 раза.

2) Это число 84 и 14.

стратегия 4: Пете прибавим 1 камень и переместим его, любого хода Вани во втором ходе имеет под-во камней больше или равное 48.

Аналогично для числа 14.

3) 

С4: Считает ~~для~~ число  $N$  Заполнит значения скоростей в массив. Подсчитает количество отрицательных эл-ов. Если оно равно 0, то найдёт максимальный отрицательный элемент и заполнит его, иначе ни вывести номера всех эл-ов  $\neq 0$ , кроме макс-го отриц-го. Иначе вывести номера всех эл-ов  $\neq 0$ .

```

Program c4;
uses crt;
var
  m, j, k, i, N: longint; but: longint;
  a: array [1..999] of longint; b: array [1..999] of longint;
begin
  k:=0; j:=1;
  readln(N);
  for i:=1 to N do
  begin
    readln(a[i]); if a[i]<0 then k:=k+1;
  end; if k mod 2=0 then begin
    for i:=1 to N do
    begin
      if a[i]<0 then b[j]:=a[i]; begin
        b[j]:=a[i];
        j:=j+1;
      end;
    end;
  end;
  for i:=1 to k-1 do
  for m:=i+1 to k do
  begin
    if a[i]< a[m] then
    begin
      but:=a[i];
      a[i]:=a[m];
      a[m]:=but;
    end;
  end;
  max:=a[1];
  for i:=1 to N do
  begin
    if (a[i]<>0) and (a[i]<max) then writeln(i);
  end;
end;
else for i:=1 to N do
  if (a[i]<>0) then writeln(i);
end;
readkey;
end.

```

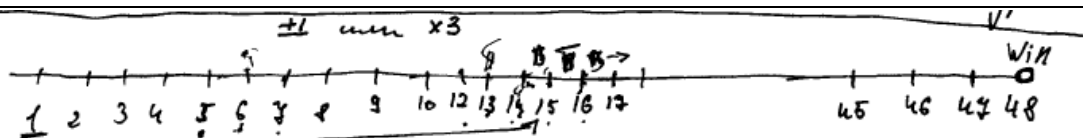
Работа 2. Вариант 4.

Оцените выполнение заданий C1-C4:

C1. 1. Программа выведет: 10.  
2. `max_digit:=10;` В данной строке взято правильное значение максимума.  
`max_digit:=0;` Лучше написать эту строку, ведь для нахождения максимума следует взять минимально возможное значение и сравнивать с ним.  
`while N>0 do` В этой строке неправильное условие выполнения цикла, при данном условии не проверяется старший разряд  
`while N<>0 do` В этом случае проверяются все разряды.

C2. `j:=0;  
min:=1000;  
for i:=1 to N do  
 if (a[i]>0) and (a[i] mod 5 = 0) and  
 (a[i] <= min) then  
 begin  
 min:=a[i];  
 j:=1;  
 end;  
if j=1 then write(min)  
else ('Не найдено');`

С3.



1. а) Петя выигрывает за один свой ход, если  $16 \leq S \leq 47$  ходом "умножить на три" однозначно. либо, если  $S = 47$  - ходом "+1". Т.к. с первого хода он увеличит число до 48 или более. При любой игре Вани.

б) Ваня выигрывает со своего ~~второго~~ <sup>первого</sup> хода, при любой игре Пети, если наимень в куче будет 15 ( $S=15$ ) (Петя остаётся только сделать первый ход "+1", после чего  $S=16$  и Ваня ходом "умножить на 3" выигрывает)

2. Петя не сможет выиграть за один ход, но сможет выиграть своим вторым ходом независимо от игры Вани, при  $S$ :

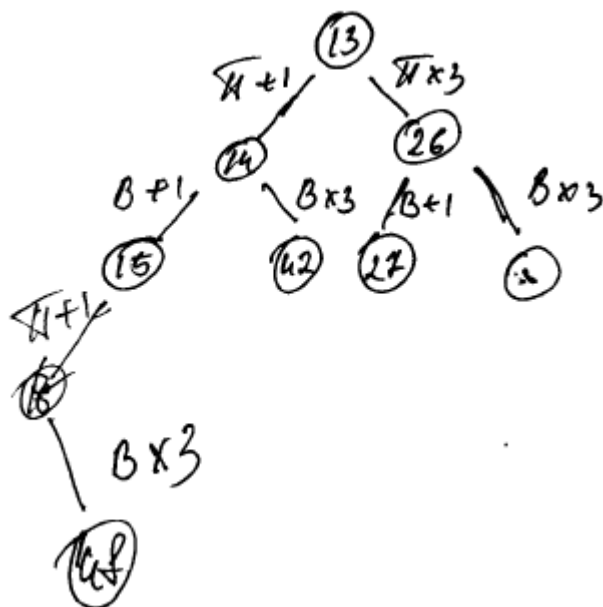
при  $S=14$  ( $14 \rightarrow$  Петя  $+1 \rightarrow 15$ ;  $15 \rightarrow$  Ваня  $+1 \rightarrow 16$ ;  $16 \rightarrow$  Петя  $\times 3 \rightarrow 48$ ) (Ваня нет смысла использовать ход "+3" т.к. заведомо проигрывает)

при  $S=5$ . ( $5 \rightarrow$  Петя  $\times 3 \rightarrow 15$ ;  $15 \rightarrow$  Ваня  $+1 \rightarrow 16$ ;  $16 \rightarrow$  Петя  $\times 3 \rightarrow 48$ ) (Петя не должен использовать ход "+1", иначе проиграет).

3. Ваня выигрывает своим вторым ходом при  $S=13$

( $13 \rightarrow$  Петя  $\rightarrow 14$ ;  $14 \rightarrow$  Ваня  $+1 \rightarrow 15$ ;  $15 \rightarrow$  Петя  $+1 \rightarrow 16$ ;  $16 \rightarrow$  Ваня  $\times 3 \rightarrow 48$ )  
переведем  $\rightarrow$

При недостатке места для ответа используйте обратную сторону бланка



C4

Введите количество элементов  $Z$  в массив введите значения скоростей. Выведите, сколько отрицательных значений. Если отрицательных значений четное число, то нам нужно вывести наибольшее отрицательное число. Если проверка, не является ли скорость равной 0, и не равна ли она наибольшему отриц. числу. Выведите на экран все порядковые номера скоростей. Программа написана на языке программирования Free Pascal

C4 (продолжение)

```
var a: array [1..N] of integer;
    max, N, i, k, j: integer;
begin
    readln(N);
    for i:=1 to N do
    a[i] := max := -109 - 1;
    k := 0;
    for i:=1 to N do begin
        readln(a[i]);
        if a[i] < 0 then begin
            if a[i] > max then j := i;
            k := k + 1;
        end;
    end;
    if k <= k div 2 then j := 0;
    for i:=1 to N do begin
        if (a[i] <= 0) and (i <= j) then write(i, ' ');
    end;
end;
```

### Работа 3. Вариант 4.

Оцените выполнение заданий C1-C4:

C1

Pascal

1. 10

2.

1) max\_digit := 10 (5 строк)

исправленное: max\_digit := 0

2) while N > 9 do (6 строк)

исправленное: while N > 0 do

$C_2$  min := 1001; (на языке паскаля)  
 for i := 1 to N do.  
 if (a[i] > 0) and (a[i] mod 5 = 0) and (a[i] < min) then min := a[i];  
 if min = 1001 then writeLn('не найдено');  
 else writeLn(min);

19. а)  $\begin{cases} S+1 \geq 48 \\ 1 \leq S \leq 47 \end{cases} \Rightarrow S = 47$   
 $\begin{cases} 3S \geq 48 \\ 1 \leq S \leq 47 \end{cases} \Rightarrow S \in [16; 47]$

При  $S \in [16; 47]$ ,  $S \in \mathbb{Z}$ , Петя может выиграть своим первым ходом. Любое число ~~от~~ из этого промежутка Петя может угадать и выигрывает, т.к. получает число  $\geq 48$ .

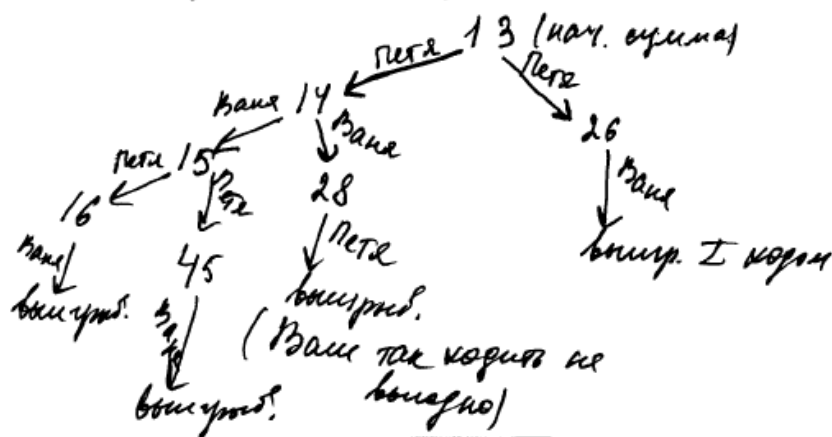
б) При  $S = 45$  Петя не может выиграть за 1 ход, но при любом его ходе, получается число или 16, или 46, следовательно, Ваня угадывает и выигрывает.

2. а) Если в куче было 3 или 14 камней, то у Пети есть выигрышная стратегия, но он не выигрывает за один ход. При  $S = 14$  Петя добавляет 1 камень в кучу и получает 15 камней и при любом ходе Вани выигрывает своим вторым ходом. При  $S = 3$  Петя может увеличить число камней в куче до 9 камней. Число не даст возможности противнику выиграть, потому будут добавлены по 1 камню в кучу, пока Ваня не сделает 16 камней, тогда Петя выигрывает своим 5 ходом.

б). Если в куче было 14 камней, то Петя увеличивает кол-во на 1, и при любом ходе Вани выигрывает своим вторым ходом.

Если в куче было 5 камней, то Петя увеличивает кол-во в три раза (становится 15 камней) и при любом ходе Вани выигрывает.

3. При  $S = 13$  Ваня выигрывает. Если Петя угадывает какое-то количество, то Ваня выигрывает своим первым ходом. Если Петя ~~не~~ увеличивает кол-во на 4, то Ваня делает то же самое и в куче становится 15 камней, следовательно, при любом ходе Пети Ваня выигрывает. Вторым ходом.



C4.

TP7.

var: a, b, c, d, e: integer.

begin.

~~readln~~

a := 0;

b := 0;

c := 0;

readln(a, b, c, d, e).

---

#### Работа 4. Вариант 4.

Оцените выполнение заданий С1-С4:

С1).

1. При вводе числа 542 программа выведет '10'. Вообще, при корректном вводе любого числа программа всегда выведет '10'. (Связано это с тем, что  $\text{max\_digit} := 10$ ; - ни одна из цифр числа не превышает данное).

2.

Ошибка а):

$\text{max\_digit} := 10$ ; Правильно:  $\text{max\_digit} := 0$ ;

Ошибка б):

while  $N \geq 0$  do Правильно: while  $N > 0$  do.

С2) Приведу фрагмент программы (Turbo Pascal 7.0):

```
...
min := 1001;
for i := 1 to N do
  if (a[i] > 0) and (a[i] mod 5 = 0) and (a[i] < min) then
    min := a[i]
  if min < 1001 then
    writeln (min)
else
  writeln ('не найдено')
...
```

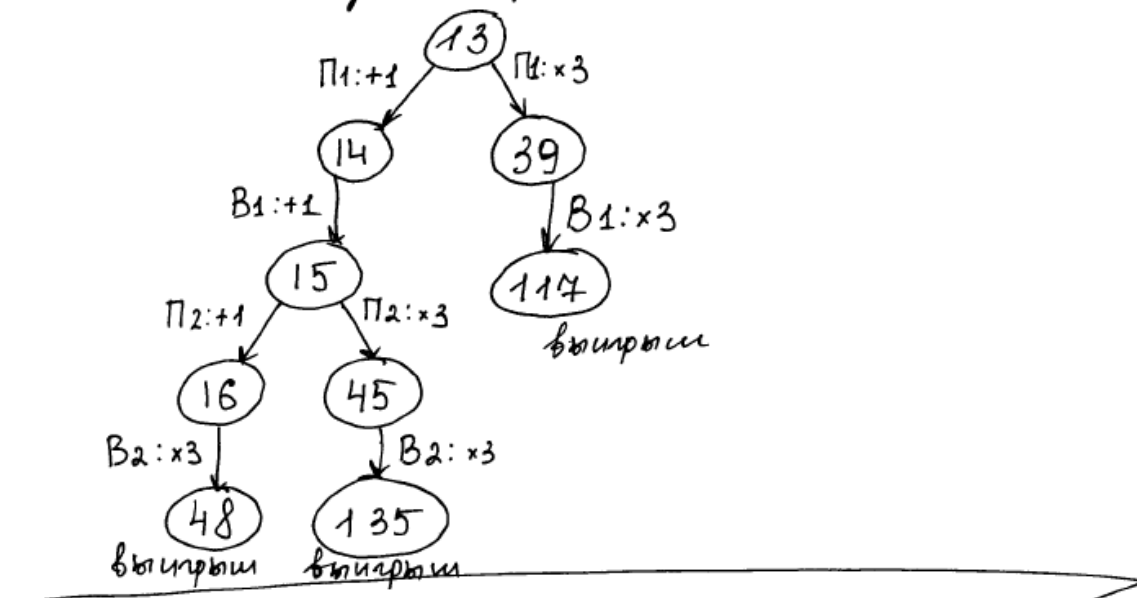


С3. 1. а) При  $16 \leq s \leq 47$  Лёте может выиграть в один ход, утроив число камней в куче. При этом, кол-во камней в куче станет не менее 48.

б) При  $s = 15$  Лёте не сможет выиграть своим первым ходом, так как при любом своем ходе Лёте приведет Ваню в выигрышную позицию (в данном случае 16 или 45), откуда Ваня может выиграть в один ход, утроив количество камней в куче.

2. При  $s = 5$  и  $s = 14$  Лёте не сможет выиграть своим первым ходом, но он может выиграть своим вторым ходом независимо от хода Вани. Для этого из позиции "5 камней" Лёте должен сделать ход "+1", а из позиции "14 камней" — "+1". При этом, после любого хода Вани, Лёте может выиграть, увеличив количество камней в куче в три раза.

3. При  $s = 13$  Ваня может выиграть своим первым или вторым ходом. Если Лёте сделает ход "+1", то Ваня сможет выиграть своим ~~первым~~ вторым ходом, сделав сначала ход "+1", а затем  $\times 3$  (т.к. независимо от хода Лёты "увеличить количество камней в три раза, число камней в куче станет не менее 48"). Если своим первым ходом Лёта увеличит количество камней в три раза, то Ваня сможет выиграть своим первым ходом, увеличив число камней в куче втрое.



C4

чтобы найти мин. произведение, я умножу  
нечетное кол-во отриц. чисел на все остальные числа  
не равные 0, при этом если кол-во <sup>отрицат.</sup> ~~нечетных~~ чисел  
четно, то необходимо отбросить наименьшее по  
модулю ~~и~~ отрицательное число

Borland Pascal 7.0

Program C4

var

N, kotr, i, minotr, imotr: integer

a: array [1..500] of integer;

begin

readln(N)

for i:=1 to N do

begin

readln(a[i]);

if (a[i]<0) then

begin

if kotr=0 then

begin

imotr:=i;

minotr:=a[i];

end

else if (a[i]>~~imotr~~ minotr) then

begin

minotr:=a[i];

```

    imotr := i;
end;
kotr kotr := kotr + 1;
end;

```

C4 (продолжение):

```

end;
if (kotr mod 2 = 1) then
    for i := 1 to N do if (a[i] <> 0) then write(i, ' ');
if (kotr mod 2 = 0) then
    for i := 1 to N do if ((a[i] <> 0) and (i <> imotr)) then
        write(i, ' ');
end.

```

#### Работа 5. Вариант 4.

Оцените выполнение заданий C1-C3:

C1) 1. 10 \*(Borland Pascal 7.0)  
 2. 1) max\_digit := 10;  
 2) max\_digit := N mod 10;

C2 min := 0;  
 (If (a[i] > 0) and If (a[i] mod 5 = 0)) then  
 min := a[i] else  
 Write ('Не найдено');  
 end;  
 Write ('Минимальное значение', min);  
 end.

С3

1. а)  $S = 16 \dots 47$  - такие значения числа  $S$  будут выигрышными для Пети, так как если взять  $S = 15$ , то получим  $15 \times 3 = 45 < 48$ .

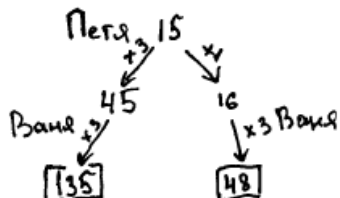
$$16 \times 3 = 48 = 48$$

...

$$47 \times 3 = 141 > 48$$

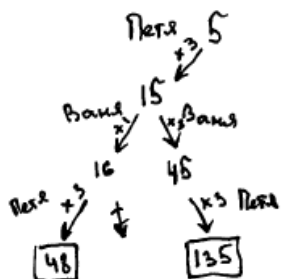
$$42 + 1 = 48 = 48$$

б)  $S = 15$  - проигрышное значение для Пети, так как если



$15 \times 3$  или  $15 + 1$  то в итоге получает число меньше 48, после чего последующий ход Вани будет выигрышным.

2.  $S = 5$

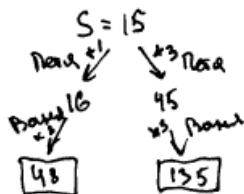


$S = 14$



$S = 5, 15$  в обоих случаях Петя не может выиграть за 1 ход, но может занять в тушке Вани, т.к. в результате получаем число  $16 \leq x < 48$ , то соответственно в следующем ходу выигрывает Петя.

3. Выигрышная стратегия



С4 - не приступал