

**Единый государственный экзамен****по ИНФОРМАТИКЕ и ИКТ****Тренировочный вариант №1802 $\alpha$** **23 апреля 2018 г.**<http://ege-inf.ru>

В экзаменационных заданиях используются следующие соглашения.

1) Обозначения для логических связок (операций):

- а) отрицание (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
- б) конъюнкция (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A\&B$ );
- в) дизъюнкция (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ ) либо  $|$  (например,  $A|B$ );
- г) следование (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
- д) тождество обозначается  $\equiv$  (например,  $A \equiv B$ ). Выражение  $A \equiv B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают (либо они оба истинны, либо они оба ложны);
- е) символ 1 используется для обозначения истины (истинного высказывания); символ 0 — для обозначения лжи (ложного высказывания).

2) Два логических выражения, содержащих переменные, называются равносильными, если значения этих выражений совпадают при любых значениях переменных. Так, выражения  $A \rightarrow B$  и  $\neg A \vee B$  равносильны, а  $A \vee B$  и  $A \wedge B$  неравносильны (значения выражений разные, например, при  $A = 1, B = 0$ ).

3) Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом,  $\neg A \wedge B \vee C \wedge D$  означает то же, что и  $((\neg A) \wedge B) \vee (C \wedge D)$ . Возможна запись  $A \wedge B \wedge C$  вместо  $(A \wedge B) \wedge C$ . То же относится и к дизъюнкции: возможна запись  $A \vee B \vee C$  вместо  $(A \vee B) \vee C$ .

4) Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле — как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

**Часть 1**

**1** Какая максимальная цифра содержится в 256-ричной записи восьмеричного числа  $27315460_8$ ? Ответ укажите в десятичной системе счисления.

**Ответ:** \_\_\_\_\_

**2** Даны логические функции  $f(x, y, z) = \neg x \wedge (y \rightarrow z)$  и  $g(x, y, z) = x \rightarrow y \wedge z$ . Сколько существует различных наборов логических переменных  $x, y, z$ , для которых истинно выражение

$$f(f(x, y, z), \neg g(z, y, x), z) \rightarrow g(y, \neg y, y) = 1?$$

**Ответ:** \_\_\_\_\_

**3** Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>A</b>	2	1	1		6	9
<b>B</b>	1		3	8		
<b>C</b>	1	3		10	3	
<b>D</b>		8	10		1	3
<b>E</b>	6		3	1		5
<b>F</b>	9			3	5	

Определите длину кратчайшего пути между пунктами А и F, **проходящего** через пункт В. Передвигаться можно только по указанным дорогам.

**Ответ:** \_\_\_\_\_

**4** Для реализации интерфейса ассоциативного массива, ключами которого являются строки, обычно используют сжатое **префиксное дерево (trie)**. Каждая дуга префиксного дерева помечена строковой, а каждая вершина — бинарной меткой. Значение ключа можно получить соединением символов, написанных на дугах на пути от корня до соответствующего выделенного узла, имеющего метку 1.

В Таблице 1 содержится информация о дугах некоторого префиксного дерева и их метках, а в Таблице 2 хранятся соответствующие метки вершин. На основании представленных данных определите количество ключей, имеющих префикс «КОД».

ID_начало	ID_конец	Метка дуги
0	1	К
1	2	А
1	3	Т
1	4	О
4	5	РА
5	6	БЛЬ
5	7	Л
9	8	ЕК
4	9	Д
8	10	С
12	11	АНИЕ
9	12	ИРОВ
12	13	КА
3	14	ИРОВКА
3	15	ЁЛ
18	16	А
18	17	ИНА
2	18	РТ
18	19	ОН
2	20	ДЕТ

ID	Метка вершины
0	0
1	0
2	0
3	1
4	0
5	0
6	1
7	1
8	1
9	1
10	1
11	1
12	0
13	1
14	1
15	1
16	1
17	1
18	0
19	1
20	1

**Ответ:** \_\_\_\_\_

**5** По каналу связи передаются сообщения, содержащие только буквы **А, Б, В, Г, Д**. Для передачи используется префиксный двоичный код. Также известно, что суммарная длина первых трёх кодовых слов равна 6.

Укажите, каким кодовым словом может быть закодирована буква **Г**. Если можно использовать более одного кодового слова, укажите кратчайшее из них, имеющее **наибольшее** числовое значение.

**Ответ:** \_\_\_\_\_

**6** Автомат получает на вход четырёхзначное десятичное число  $N$ . По этому числу строится новое число по следующим правилам.

- 1) Строятся числа  $N_{12}$ ,  $N_{34}$ , которые образованы соответственно двумя левыми и двумя правыми цифрами числа  $N$  (в числах  $N_{12}$ ,  $N_{34}$  цифры идут в том же порядке, что и в числе  $N$ ).
- 2) Вычисляется и печатается в качестве результата поразрядная конъюнкция  $N_{12} \& N_{34}$ .

**Пример.** Исходное число:  $N = 1206$ .  $N_{12} = 12$ ;  $N_{34} = 06$ .  $N_{12} \& N_{34} = 12 \& 6 = 4$ .  
**Результат:** 4.

Укажите количество различных чисел, при обработке которых автомат выдаёт результат **50**.

**Ответ:** \_\_\_\_\_

**7** Для вычисления биномиальных коэффициентов  $C_n^k$  во все ячейки столбца **А** и строки **1** электронной таблицы записали единицы, а в ячейку **В2** — формулу **=В1+А2**, как показано на рисунке.

	<b>А</b>	<b>В</b>	<b>С</b>		<b>Е</b>
<b>1</b>	1	1	1		1
<b>2</b>	1	=В1+А2			
<b>3</b>	1				
<b>7</b>	1				

Чему станет равно значение в ячейке **Е7**, если формулу из ячейки **В2** скопировать во все ячейки диапазона **В2:Е7**?

**Ответ:** \_\_\_\_\_

**8** Запишите число, которое будет напечатано в результате выполнения следующей программы.

Pascal	C++
<pre> <b>var</b> p, i : integer; <b>begin</b>   p := 50; i := 0;   <b>while</b> p &gt;= 0 <b>do begin</b>     i := i - 3 + p <b>mod</b> 5;     p := p - 1   <b>end;</b>   write(i) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>int</b> main() {   <b>int</b> p=50, i=0;   <b>while</b> (p &gt;= 0) {     i = i - 3 + p % 5;     p--;   }   std::cout &lt;&lt; i; } </pre>
Basic	Python
<pre> <b>DIM</b> P, I <b>AS</b> INTEGER P = 50 : I = 0 <b>WHILE</b> P &gt;= 0   I = I - 3 + P <b>MOD</b> 5 : P = P - 1 <b>WEND</b> <b>PRINT</b> I </pre>	<pre> p, i = 50, 0 <b>while</b> p &gt;= 0 :   i = i - 3 + p % 5   p -= 1 <b>print</b>(i) </pre>

**Ответ:** \_\_\_\_\_

**9** Известно, что объём некоторого растрового изображения размером  $640 \times 512$  пикселей на 80 Кбайт меньше объёма некоторой одноканальной (моно) звукозаписи с частотой дискретизации 16 кГц и длительностью 7,68 секунды. Если количество цветов, используемых в палитре этого изображения, увеличить в 256 раз, а разрешение звукозаписи увеличить в 2 раза, то объём изображения станет равен объёму звукозаписи. Определите первоначальное разрешение звукозаписи. Ответ дайте в битах, сжатие во всех случаях не производилось.

**Ответ:** \_\_\_\_\_

**10** Для передачи сообщений используются кодовые слова, содержащие ровно три буквы *A*, две буквы *B* и четыре буквы *C*, записанные в произвольном порядке. Укажите количество возможных кодовых слов.

**Ответ:** \_\_\_\_\_

**11** Ниже на четырёх языках записан рекурсивный алгоритм F.

Pascal	C++
<pre> <b>function</b> F(n : integer) : integer; <b>begin</b>   <b>if</b> n &gt; 2 <b>then</b>     F := -6 * F(n - 1) - 12 * F(n - 2) -           8 * F(n - 3)   <b>else</b>     F := n + 1 <b>end;</b> </pre>	<pre> <b>int</b> F(<b>int</b> n) {   <b>if</b> (n &gt; 2)     <b>return</b> -6 * F(n - 1) - 12 * F(n - 2) -            8 * F(n - 3);   <b>else</b>     <b>return</b> n + 1; } </pre>
Basic	Python
<pre> <b>FUNCTION</b> F(N)   <b>IF</b> N &gt; 2 <b>THEN</b>     F = -6 * F(N - 1) - 12 * F(N - 2) -         8 * F(N - 3)   <b>ELSE</b>     F = N + 1   <b>END IF</b> <b>END FUNCTION</b> </pre>	<pre> <b>def</b> F(n) :   <b>if</b> n &gt; 2 :     <b>return</b> -6 * F(n - 1) - 12 * F(n - 2) - \            8 * F(n - 3)   <b>else</b> :     <b>return</b> n + 1 </pre>

Чему будет равно значение, вычисленное при выполнении вызова F(8)?

**Ответ:** \_\_\_\_\_

**12** Компьютер имеет порядковый номер 4000 в сети с адресом 80.90.144.0 и маской 255.255.240.0. Укажите третий байт IP адреса этого компьютера. Ответ дайте в десятичной системе счисления.

**Ответ:** \_\_\_\_\_

**13** При регистрации в системе пользователю выдаётся ключ, содержащий только строчные буквы латинского алфавита. Для хранения каждого ключа отведено одинаковое минимально возможное целое число байт. При этом используют посимвольное кодирование ключей, все символы кодируют одинаковым минимально возможным количеством бит.

Для хранения 200 ключей потребовалось 800 байт. При расширении системы было решено увеличить длины всех ключей на 2 символа, что увеличило объём данных на 25%. Какова минимальная длина (в символах) одного старого ключа?

**Ответ:** \_\_\_\_\_

**14** Исполнитель 3D чертёжник перемещается в пространстве, оставляя след в виде линии. 3D чертёжник может выполнять команду сместиться на  $(a, b, c)$ , где  $a, b, c$  — целые числа, которая перемещает 3D чертёжника из точки с координатами  $(x, y, z)$  в точку с координатами  $(x + a, y + b, z + c)$ .

Цикл

```
ПОВТОРИ число РАЗ
    последовательность команд
КОНЕЦ ПОВТОРИ
```

означает, что последовательность команд будет выполнена указанное число раз (число должно быть натуральным).

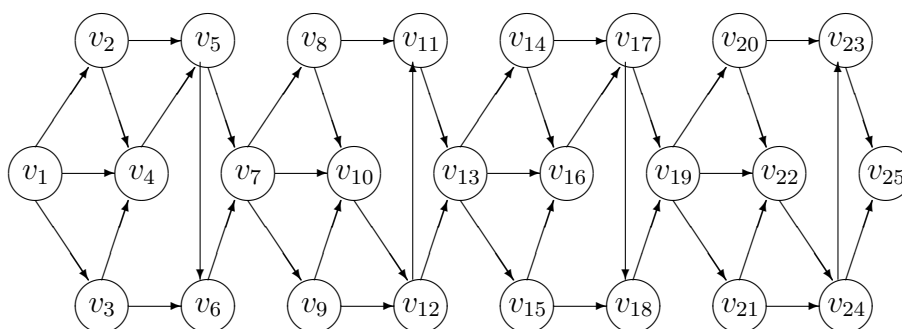
3D чертёжнику был дан для исполнения следующий алгоритм:

```
НАЧАЛО
    сместиться на (0, -28, -140)
    ПОВТОРИ n РАЗ
        сместиться на (a, -b, 2c)
        сместиться на (-123456, 654321, 123456)
    КОНЕЦ ПОВТОРИ
    сместиться на (35, -98, 20)
КОНЕЦ
```

Укажите **наибольшее** значение числа  $n$ , для которого найдутся такие значения чисел  $a, b$  и  $c$ , что после выполнения программы 3D чертёжник возвратится в точку, которая находится на расстоянии **не более 1** от исходной точки.

**Ответ:** \_\_\_\_\_

**15** Схема дорог, связывающая 25 городов, показана на рисунке. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Во сколько раз уменьшится количество различных путей из города  $v_1$  в город  $v_{25}$ , если из схемы дорог исключить город  $v_{16}$  со всеми инцидентными ему дорогами?



**Ответ:** \_\_\_\_\_

**16** Решите в **натуральных** числах уравнение

$$2^{3x} + 4^{2x-2y} + 2^{5y} = 548.$$

В ответе укажите сумму  $x + y$  в **двоичной** системе счисления.

**Ответ:** \_\_\_\_\_

**17** В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» — символ «&». В таблице приведены запросы и количество найденных по ним страниц.

Запрос	Количество страниц
<i>Язык</i>	1000
<i>Фортран</i>	100
<i>Язык &amp; Бейсик</i>	200
<i>Язык &amp; Бейсик &amp; Фортран</i>	50
<i>Язык   Бейсик   Фортран</i>	1000

Какое количество страниц будет найдено по запросу *Бейсик | Фортран*?

**Ответ:** \_\_\_\_\_

**18** Для какого **наименьшего** целого числа  $A$  формула

$$(3x + 4y + 8z \leq 24) \rightarrow \left( (x - 4)^2 + (y - 3)^2 + z^2 < A \right)$$

тождественно **истинна** для любых неотрицательных чисел  $x, y, z$ ?

**Ответ:** \_\_\_\_\_



**19** В программе используется трёхмерный целочисленный массив  $A$  с индексами от 0 до 1, от 0 до 2 и от 0 до 3 соответственно. Определите значение переменной  $s$  после выполнения следующего фрагмента программы.

Pascal	C++
<pre> s := 0; <b>for</b> i := 0 <b>to</b> 1 <b>do</b>   <b>for</b> j := 0 <b>to</b> 2 <b>do</b>     <b>for</b> k := 0 <b>to</b> 3 <b>do</b>       A[i, j, k] := i + j + k + 1; <b>for</b> t := 0 <b>to</b> 15 <b>do begin</b>   x := A[t <b>mod</b> 2, t <b>mod</b> 3, t <b>mod</b> 4];   A[t <b>mod</b> 2, t <b>mod</b> 3, t <b>mod</b> 4] := t <b>div</b> x + 1;   s := s + t <b>mod</b> x <b>end;</b> </pre>	<pre> s = 0; <b>for</b> (i = 0; i &lt; 2; i++)   <b>for</b> (j = 0; j &lt; 3; j++)     <b>for</b> (k = 0; k &lt; 4; k++)       A[i][j][k] = i + j + k + 1; <b>for</b> (t = 0; t &lt; 16; t++) {   x = A[t % 2][t % 3][t % 4];   A[t % 2][t % 3][t % 4] = t / x + 1;   s += t % x; } </pre>
Basic	Python
<pre> s = 0 <b>For</b> i = 0 <b>To</b> 1   <b>For</b> j = 0 <b>To</b> 2     <b>For</b> k = 0 <b>To</b> 3       A(i, j, k) = i + j + k + 1 <b>Next</b> k : <b>Next</b> j : <b>Next</b> i <b>For</b> t = 0 <b>To</b> 15   x = A(t <b>Mod</b> 2, t <b>Mod</b> 3, t <b>Mod</b> 4)   A(t <b>Mod</b> 2, t <b>Mod</b> 3, t <b>Mod</b> 4) = t \ x + 1   s = s + t <b>Mod</b> x <b>Next</b> t </pre>	<pre> s = 0 <b>for</b> i <b>in</b> range(2) :   <b>for</b> j <b>in</b> range(3) :     <b>for</b> k <b>in</b> range(4) :       A[i][j][k] = i + j + k + 1 <b>for</b> t <b>in</b> range(16) :   x = A[t % 2][t % 3][t % 4]   A[t % 2][t % 3][t % 4] = t // x + 1   s += t % x </pre>

**Ответ:** \_\_\_\_\_

**20** Укажите **наибольшее** натуральное число  $x$ , при вводе которого будут напечатаны числа 8 и 5.

Pascal	C++
<pre> <b>var</b> x, a, b : integer; <b>begin</b>   read(x);   a := 0; b := 0;   <b>while</b> x &gt; 1 <b>do begin</b>     a := a + 1;     <b>if</b> x mod 2 &lt;&gt; x div 2 mod 2 <b>then</b>       b := b + 1;     x := x div 2   <b>end;</b>   writeln(a, ' ', b) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>int</b> main() {   <b>int</b> x, a = 0, b = 0;   std::cin &gt;&gt; x;   <b>while</b> (x &gt; 1) {     a++;     <b>if</b> (x % 2 != x / 2 % 2)       b++;     x /= 2;   }   std::cout &lt;&lt; a &lt;&lt; " " &lt;&lt; b; } </pre>
Basic	Python
<pre> <b>DIM</b> X, A, B <b>AS</b> INTEGER INPUT X A = 0 : B = 0 <b>WHILE</b> X &gt; 1   A = A + 1   <b>IF</b> X MOD 2 &lt;&gt; X \ 2 MOD 2 <b>THEN</b>     B = B + 1   <b>ENDIF</b>   X = X \ 2 <b>WEND</b> PRINT A, " ", B </pre>	<pre> x = int(input()) a = b = 0 <b>while</b> x &gt; 1 :   a += 1   <b>if</b> x % 2 != x // 2 % 2 :     b += 1   x //= 2 print(a, b) </pre>

**Ответ:** \_\_\_\_\_

**21** Определите, какое число будет напечатано в результате выполнения следующего алгоритма.

Pascal	C++
<pre> <b>function</b> f(x: integer) : integer; <b>begin</b>   <b>if</b> x &gt; 2 <b>then</b>     f := x * x - 23   <b>else</b>     f := x * x + 3 * x - 7 <b>end;</b>  <b>var</b>   t, M: integer; <b>begin</b>   M := -5;   <b>for</b> t := -4 <b>to</b> 5 <b>do</b>     <b>if</b> f(t) &gt; f(M) <b>then</b>       M := t;   writeln(5 - M) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt;  <b>int</b> f(<b>int</b> x) {   <b>if</b> (x &gt; 2)     <b>return</b> x * x - 23;   <b>else</b>     <b>return</b> x * x + 3 * x - 7; }  <b>int</b> main() {   <b>int</b> M = -5;   <b>for</b> (<b>int</b> t = -4; t &lt; 6; t++)     <b>if</b> (f(t) &gt; f(M))       M = t;   std::cout &lt;&lt; 5 - M;   <b>return</b> 0; } </pre>
Basic	Python
<pre> <b>DIM</b> T, M <b>AS</b> <b>INTEGER</b> M = -5 <b>FOR</b> T = -4 <b>TO</b> 5   <b>IF</b> F(T) &gt; F(M) <b>THEN</b>     M = T   <b>ENDIF</b> <b>NEXT</b> T <b>PRINT</b> 5 - M  <b>FUNCTION</b> F(X)   <b>IF</b> F(T) &gt; F(M) <b>THEN</b>     F = X * X - 23   <b>ELSE</b>     F = X * X + 3 * X - 7   <b>ENDIF</b> <b>END FUNCTION</b> </pre>	<pre> <b>def</b> f(x) :   <b>if</b> x &gt; 2 :     <b>return</b> x * x - 23   <b>else</b> :     <b>return</b> x * x + 3 * x - 7  M = -5 <b>for</b> t <b>in</b> range(-4, 6) :   <b>if</b> f(t) &gt; f(M) :     M = t <b>print</b>(5 - M) </pre>

Ответ: \_\_\_\_\_

**22** Для некоторого исполнителя, преобразующего число на экране, определены две команды:

- 1) прибавить 1;
- 2) умножить на 2.

Первая из них увеличивает число на экране на 1, вторая увеличивает это число в 2 раза. Программа для исполнителя — это последовательность команд.

Сколько существует программ для этого исполнителя, которые содержат **не менее восьми команд** и преобразуют число **8** в число **42**?

**Ответ:** \_\_\_\_\_

**23** Сколько существует различных наборов значений логических переменных  $x_1, \dots, x_4, y_1, \dots, y_4$  при которых следующие выражения истинны?

$$\begin{aligned}(\neg x_1 \equiv y_1) &\rightarrow (x_2 \equiv \neg y_2), \\(x_1 \equiv \neg y_1) &\rightarrow (x_2 \equiv \neg y_2), \\(\neg x_2 \equiv y_2) &\rightarrow (x_3 \equiv \neg y_3), \\(\neg x_2 \equiv y_2) &\rightarrow (\neg x_3 \equiv y_3), \\(\neg x_3 \equiv y_3) &\rightarrow (x_4 \equiv \neg y_4), \\(x_3 \equiv \neg y_3) &\rightarrow (\neg x_4 \equiv y_4), \\(\neg x_1 \equiv x_4) &\rightarrow (y_4 \equiv \neg y_1).\end{aligned}$$

**Ответ:** \_\_\_\_\_

## Часть 2

**24** Требовалось написать программу, которая по входному натуральному числу  $n$ , не превосходящему  $10^9$ , определяла и печатала номер позиции первой входящей в него цифры 4, считая слева направо. Если цифры 4 в числе  $n$  нет, необходимо вывести 0. Приведённая ниже программа содержит ошибки.

Pascal	C++
<pre> <b>var</b>   j, k, n : longint; <b>begin</b>   read(n);   k := 0;   j := 0;   <b>while</b> n &gt; 0 <b>do begin</b>     <b>if</b> n mod 10 = 4 <b>then</b>       j := k;       k := k + 1;       n := n div 10   <b>end;</b>   <b>if</b> j &gt; 0 <b>then</b>     write(k - 1)   <b>else</b>     write(j) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>void</b> main() {   <b>int</b> n, j, k;   std::cin &gt;&gt; n;   k = 0;   j = 0;   <b>while</b> (n &gt; 0) {     <b>if</b> (n % 10 == 4)       j = k;       k++;       n /= 10;   }   <b>if</b> (j &gt; 0)     std::cout &lt;&lt; k - 1;   <b>else</b>     std::cout &lt;&lt; j; } </pre>
Basic	Python
<pre> <b>Dim</b> n, j, k <b>As</b> Long Input n k = 0 j = 0 <b>While</b> n &gt; 0   <b>If</b> n Mod 10 = 4 <b>Then</b>     j = k   <b>End If</b>   k = k + 1   n = n \ 10 <b>Wend</b> <b>If</b> j &gt; 0 <b>Then</b>   Print k - 1 <b>Else</b>   Print j <b>End If</b> </pre>	<pre> n = int(input()) k = 0 j = 0 <b>while</b> n &gt; 0 :   <b>if</b> n % 10 == 4 :     j = k     k += 1     n //= 10 <b>if</b> j &gt; 0 :   print(k - 1) <b>else</b> :   print(j) </pre>

Последовательно выполните следующее.

- 1) Напишите, что выведет эта программа при вводе числа **1474**.
- 2) Укажите **наибольшее трёхзначное** значение входной переменной  $n$ , при котором, не смотря на ошибки, программа выведет верный ответ, **не равный нулю**.
- 3) Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Достаточно указать ошибки и способ их исправления для одного языка программирования. Обратите внимание: необходимо исправить приведённую программу, а не написать свою. Требуется заменить только ошибочные строки, но нельзя удалять строки или добавлять новые. За исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

**25** Дан массив  $A$ , содержащий 2018 натуральных чисел из отрезка  $[10; 999]$ . Необходимо определить количество элементов этого массива, в записи которых **вторая слева** цифра **кратна трём**. Например, в массиве из 4-х элементов, равных соответственно 30; 31; 987; 161, имеется два числа — 30 и 161, у которых вторая слева цифра кратна трём.

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

Pascal	C++
<pre> <b>const</b>   N=2018; <b>var</b>   a : <b>array</b> [0..N-1] <b>of</b> integer;   i, m : integer; <b>begin</b>   <b>for</b> i:=0 <b>to</b> N-1 <b>do</b>     readln(a[i]);     ... <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; #define N 2018 <b>int</b> main() {   <b>int</b> a[N], i, m;   <b>for</b> (i=0; i&lt;N; i++)     std::cin&gt;&gt;a[i];   ...   <b>return</b> 0; } </pre>

Basic	Python
<pre>CONST N=2018 DIM A(N-1), I, M AS INTEGER FOR I=0 TO N-1     INPUT A(I) NEXT I ... END</pre>	<pre># допускается также использование # целочисленной переменной m N=2018 a = [] for i in range(N):     a.append(int(input())) ...</pre>

В качестве ответа необходимо привести фрагмент программы, который должен находиться на месте многоточия. Допускается решение на другом языке программирования. В этом случае укажите название языка и используемую версию. При этом необходимо использовать те же самые исходные данные и переменные, какие были предложены в условии.

**26** Три игрока, Паша, Вова и Тимур, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша, второй ход — Вова и третий ход — Тимур соответственно. За один ход игрок может добавить в кучу **семь** камней или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 22 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится **не менее** 40. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 40 или больше камней. В начальный момент в куче было  $1 \leq S \leq 39$  камней.

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) Укажите все такие значения числа  $S$ , при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход и может выиграть своим вторым ходом независимо от того, как будут ходить Вова и Тимур. Для каждого указанного значения  $S$  опишите выигрышную стратегию Паши.
- 2) Известно, что Паша и Вова создали коалицию и играют согласованно. Укажите все такие значения числа  $S$ , при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход и может выиграть своим вторым ходом независимо от того, как будет ходить Тимур. Для каждого указанного значения  $S$  опишите выигрышную стратегию Паши.
- 3) Известно, что Паша и Тимур создали коалицию и играют согласованно. Укажите все такие значения числа  $S$ , при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход и может выиграть своим вторым ходом независимо от того, как будет ходить Вова. Для наибольшего из указанных значений  $S$  опишите выигрышную стратегию Паши. Постройте дерево всех партий, возможных при этой выигрышной стратегии Паши (в виде рисунка или таблицы).



**27** Дан набор из  $N$  натуральных чисел. Необходимо определить количество троек элементов  $(a_i, a_j, a_k)$  этого набора, в которых  $1 \leq i < j < k \leq N$  и **сумма** элементов **кратна 12**.

Напишите эффективную по времени и по памяти программу для решения этой задачи. Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз. Программа считается эффективной по памяти, если память, необходимая для хранения переменных программы, не превышает одного килобайта и не увеличивается с ростом  $N$ .

#### **Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел  $N$  ( $3 \leq N \leq 10000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 1000.

*Пример входных данных:*

5  
7  
5  
6  
12  
24

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

2

В приведённом наборе из 5 чисел имеются две тройки —  $(7, 5, 12)$  и  $(7, 5, 24)$ , сумма элементов которых кратна 12.