

**Единый государственный экзамен****по ИНФОРМАТИКЕ и ИКТ****Тренировочный вариант №1801****22 декабря 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** Сколько значащих нулей содержится в **четверичной** записи шестнадцатеричного числа  $123ABC0_{16}$ ?

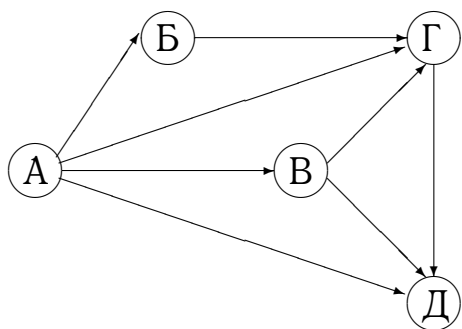
**Ответ:** \_\_\_\_\_

**2** Укажите количество **различных** логических функций трёх логических переменных  $x, y, z$ , которые удовлетворяют приведённому ниже фрагменту таблицы истинности.

$x$	$y$	$z$	$f(x, y, z)$
1	0	1	1
0	0	1	0

**Ответ:** \_\_\_\_\_

**3** На рисунке схема дорог некоторого района изображена в виде графа, в таблице содержатся сведения о длине этих дорог.



	<b>П1</b>	<b>П2</b>	<b>П3</b>	<b>П4</b>	<b>П5</b>
<b>П1</b>					1
<b>П2</b>	1		1	4	3
<b>П3</b>				3	2
<b>П4</b>					
<b>П5</b>				1	

Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите длину самого протяжённого маршрута от пункта А до пункта Г.

**Ответ:** \_\_\_\_\_

**4** При проектировании табличных баз данных для хранения и обработки иерархических структур может использоваться шаблон *Closure Table* («таблица связей»). Согласно этому шаблону, структура страниц некоторого web-сайта хранится в двух таблицах. В таблице *pages* содержатся данные о страницах, а в таблице *pages\_tree-path* — связи между ними. Таблица связей содержит два поля: ссылку на предка (*ancestor*) и ссылку на потомка (*descendant*).

По данным таблиц *pages* и *pages\_tree-path* определите количество страниц web-сайта, имеющих первый уровень вложенности.

Таблица <i>pages</i>	
id	title
1	О компании
2	Цены
3	Контакты
4	Вакансии
5	Менеджер по рекламе
6	Веб-дизайнер
7	Сайты
8	Логотипы
9	Знаки
10	Визитка
11	Корпоративный

Таблица <i>pages_tree-path</i>	
ancestor	descendant
1	3
1	4
1	5
1	6
4	5
4	6
2	7
2	10
2	11
2	8
2	9
7	10
7	11

**Ответ:** \_\_\_\_\_

**5** Некоторое сообщение содержит только буквы А, Б, В, Г, Д, причём известно их количество: А — 179, Б — 89, В — 72, Г — 53 и Д — 50. Сколько бит содержит оптимальный префиксный код данного сообщения?

**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 **наибольшее** число, при обработке которого автомат выдаёт результат **50**.

**Ответ:** \_\_\_\_\_

**7** Ячейки диапазона **A1:B2** электронной таблицы имеют значения, как показано на рисунке ниже.

	<b>A</b>	<b>B</b>
<b>1</b>	1	1
<b>2</b>	=\$A1*2	=\$B1+A2
<b>3</b>		
...		
<b>10</b>		

Чему станет равно значение в ячейке **B8**, если формулы из ячеек **A2** и **B2** скопировать во все ячейки диапазонов **A3:A10** и **B3:B10** соответственно?

**Ответ:** \_\_\_\_\_

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

Паскаль	C++
<pre> <b>var</b> p, i : integer; <b>begin</b>   p := 1; i := 4;   <b>while</b> i &lt;= 14 <b>do begin</b>     p := p * (i <b>div</b> 4);     i := i + 1   <b>end;</b>   write(p) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>int</b> main() {   <b>int</b> p=1, i=4;   <b>while</b> (i &lt;= 14) {     p *= i / 4;     i++;   }   std::cout &lt;&lt; p; } </pre>
Бейсик	Python
<pre> <b>Dim</b> P, I <b>As</b> Integer P=1 : I=4 <b>While</b> I &lt;= 14   P=P*(I\4) : I=I+1 <b>End While</b> Print P </pre>	<pre> p, i = 1, 4 <b>while</b> i &lt;= 14 :   p *= i // 4   i += 1 print(p) </pre>

**Ответ:** \_\_\_\_\_

**9** Камера видеорегистратора имеет скорость записи 25 кадров/сек. и разрешение Full HD (1920x1080 пикселей) в режиме TrueColor (24 бита/пиксель). Также регистратор оснащён жёстким диском объёмом 250 Гбайт. Укажите максимальную продолжительность видео (в минутах), которое может снять регистратор, если сжатие данных не используется. Ответ округлите до 10 минут.

**Ответ:** \_\_\_\_\_

**10** Для передачи сообщений используются 5-буквенные слова, в которых есть только буквы А, В, С, D, причём каждая из букв А, В, С может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько раз в слове должна встречаться буква D, если известно, что всего имеется 405 различных кодовых слов?

**Ответ:** \_\_\_\_\_

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

Паскаль	C++
<pre> <b>function</b> F(n : integer) : integer; <b>begin</b>   <b>if</b> n &gt; 2 <b>then</b>     F := F(n - 1) - F(n - 2)   <b>else</b>     <b>if</b> n = 2 <b>then</b>       F := F(n - 1) * F(n - 1)     <b>else</b>       F := 3     <b>end if</b>   <b>end if</b> <b>end;</b> </pre>	<pre> <b>int</b> F(<b>int</b> n) {   <b>if</b> (n &gt; 2)     <b>return</b> F(n - 1) - F(n - 2);   <b>else</b>     <b>if</b> (n == 2)       <b>return</b> F(n - 1) * F(n - 1);     <b>else</b>       <b>return</b> 3;   } </pre>
Бейсик	Python
<pre> <b>Function</b> F(N)   <b>If</b> N &gt; 2 <b>Then</b>     F = F(N - 1) - F(N - 2)   <b>Else</b>     <b>If</b> N = 2 <b>Then</b>       F = F(N - 1) * F(N - 1)     <b>Else</b>       F = 3     <b>End If</b>   <b>End If</b> <b>End Function</b> </pre>	<pre> <b>def</b> F(n) :   <b>if</b> n &gt; 2 :     <b>return</b> F(n - 1) - F(n - 2)   <b>else</b> :     <b>if</b> n == 2 :       <b>return</b> F(n - 1) * F(n - 1)     <b>else</b> :       <b>return</b> 3 </pre>

Чему будет равно значение, вычисленное при выполнении вызова F(200)?

**Ответ:** \_\_\_\_\_

**12** Для узла с IP-адресом 111.3.129.27 адрес сети равен 111.3.128.0. Сколько существует возможных значений третьего слева байта маски?

**Ответ:** \_\_\_\_\_



**16** Сколько нечётных цифр в восьмеричной записи числа  $4^{10} - 2^{10} + \frac{1}{2} \cdot 8^2 - 16^2$ ?

**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>пролог</i>	100
<i>пролог   алгоритм</i>	150
<i>пролог   паскаль</i>	200
<i>алгоритм &amp; паскаль</i>	30
<i>паскаль</i>	100

Какое количество страниц будет найдено по запросу

*пролог | алгоритм | паскаль*?

**Ответ:** \_\_\_\_\_

**18** На вещественной плоскости  $\mathbb{R}^2$  заданы две области:  $P(x, y) = \{y < |x|\}$  и  $Q(x, y) = \{y > \sqrt{4\pi^{-1} - x^2}\}$ . Укажите **наибольшую** возможную площадь области  $A$ , такой, что формула

$$((x, y) \in A) \wedge (((x, y) \in Q) \vee ((x, y) \in P))$$

тождественно **ложна** для любой пары чисел  $(x, y) \in \mathbb{R}^2$ .

**Ответ:** \_\_\_\_\_



**19** В программе используется одномерный целочисленный массив  $A$  с индексами от 0 до 20. Известно, что в массиве хранятся натуральные двузначные числа. Определите **наибольшее** возможное значение переменной  $k$  после выполнения следующего фрагмента программы.

Паскаль	C++
<pre>k := 0; for i := 0 to 20 do   k := k + (3 - i mod 6) * A[i];</pre>	<pre>k = 0; for (i = 0; i &lt;= 20; i++)   k += (3 - i % 6) * A[i];</pre>
Бейсик	Python
<pre>K = 0 For I = 0 To 20   K = K + (3 - I Mod 6) * A(I) Next I</pre>	<pre>k = 0 for i in range(21):   k += (3 - i % 6) * A[i]</pre>

**Ответ:** \_\_\_\_\_

**20** Укажите **наименьшее** натуральное число  $x$ , при вводе которого будут напечатаны числа 5 и 6.

Паскаль	C++
<pre> <b>var</b> x, a, b : integer; <b>begin</b>   read(x);   a := 0;   b := 0;   <b>while</b> x &gt; 0 <b>do begin</b>     a := a + 1;     <b>if</b> x mod 10 mod a = 1 <b>then</b>       b := b + a;     x := x div 10   <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; 0) {     a++;     <b>if</b> (x % 10 % a == 1)       b += a;     x /= 10;   }   std::cout &lt;&lt; a &lt;&lt; " " &lt;&lt; b;   <b>return</b> 0; } </pre>
Бейсик	Python
<pre> <b>Dim</b> X, A, B <b>As</b> Integer Input X A = 0 : B = 0 <b>While</b> X &gt; 0   A = A + 1   <b>If</b> B Mod 10 Mod A = 1 <b>Then</b>     B = B + A   <b>End If</b>   X = X \ 10 <b>End While</b> Print A, " ", B </pre>	<pre> x = int(input()) a = b = 0 <b>while</b> x &gt; 0 :   a += 1   <b>if</b> x % 10 % a == 1 :     b += a   x //= 10 print(a, b) </pre>

**Ответ:** \_\_\_\_\_

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

Паскаль	C++
<pre> <b>function</b> f(x : integer) : integer; <b>begin</b>   f := x * x <b>end;</b>  <b>function</b> g(x : integer) : integer; <b>begin</b>   g := 2 * x + 1 <b>end;</b>  <b>var</b>   t : integer; <b>begin</b>   t := f(4);   <b>while</b> f(g(t)) &lt;&gt; g(f(t)) <b>do</b>     t := t - 3;     writeln(t) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt;  <b>int</b> f(<b>int</b> x) {   <b>return</b> x * x; }  <b>int</b> g(<b>int</b> x) {   <b>return</b> 2 * x + 1; }  <b>int</b> main() {   <b>int</b> t = f(4);   <b>while</b> (f(g(t)) != g(f(t)))     t -= 3;   std::cout &lt;&lt; t; } </pre>
Бейсик	Python
<pre> <b>Dim</b> T <b>As</b> Integer T = F(4) <b>While</b> F(G(T)) &lt;&gt; G(F(T))   T = T - 3 <b>End While</b> Print T  <b>Function</b> F(X)   F = X * X <b>End Function</b>  <b>Function</b> G(X)   G = 2 * X + 1 <b>End Function</b> </pre>	<pre> <b>def</b> f(x) :   <b>return</b> x * x  <b>def</b> g(x) :   <b>return</b> 2 * x + 1  t = f(4) <b>while</b> f(g(t)) != g(f(t)) :   t -= 3 print(t) </pre>

**Ответ:** \_\_\_\_\_

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

1) прибавить 1;

2) прибавить 2;

3) прибавить 3.

Каждая из них увеличивает число на экране на 1, 2 и 3 соответственно. Программа для исполнителя — это последовательность команд.

Сколько существует программ, для которых при исходном числе **2** результатом является число **15**, при этом траектория вычислений содержит **все** числа отрезка  $[2; 15]$ , **кратные 4**?

**Ответ:** \_\_\_\_\_

**23** Сколько существует различных наборов значений логических переменных  $x_1, \dots, x_{10}$ , при которых следующие выражения истинны?

$$((x_1 \rightarrow x_2) \vee x_1 \wedge \neg x_2) \wedge (x_3 \vee x_4),$$

$$((x_3 \rightarrow x_4) \vee x_3 \wedge \neg x_4) \wedge (x_5 \vee x_6),$$

$$((x_5 \rightarrow x_6) \vee x_5 \wedge \neg x_6) \wedge (x_7 \vee x_8),$$

$$((x_7 \rightarrow x_8) \vee x_7 \wedge \neg x_8) \wedge (x_9 \vee x_{10}).$$

**Ответ:** \_\_\_\_\_

## Часть 2

**24** Требовалось написать программу, которая вычисляет значение многочлена

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

при заданном значении аргумента  $x$ . На вход последовательно подаются неотрицательная целая степень многочлена  $n$ , значение аргумента  $x$  и  $n + 1$  значение коэффициентов многочлена  $a_n, \dots, a_0$ . Приведённая ниже программа содержит ошибки.

Паскаль	C++
<pre> <b>var</b>   n, x, a, p, i : integer; <b>begin</b>   read(n, x);   p := x;   <b>for</b> i := n <b>downto</b> 1 <b>do begin</b>     read(a);     p := p * x + a   <b>end;</b>   write(p) <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>int</b> main() {   <b>int</b> n, x, a;   std::cin &gt;&gt; n &gt;&gt; x;   <b>int</b> p = x;   <b>for</b> (<b>int</b> i = n; i &gt;= 1; i--) {     std::cin &gt;&gt; a;     p = p * x + a;   }   std::cout &lt;&lt; p;   <b>return</b> 0; } </pre>
Бейсик	Python
<pre> <b>Dim</b> N, X, A, P, I <b>As</b> Integer Inout N, X P = X <b>For</b> I = N <b>To</b> 1 <b>Step</b> -1   Input A   P = P * X + A <b>Next</b> I Print P </pre>	<pre> n = int(input()) x = int(input()) p = x <b>for</b> i <b>in</b> range(n, 0, -1) :   a = int(input())   p = p * x + a print(p) </pre>

Последовательно выполните следующее.

- 1) Напишите, что выведет эта программа при вводе чисел 2 1 2 1 0.
- 2) Приведите пример входных данных, при которых, не смотря на ошибки, программа выведет верный ответ и укажите этот ответ.
- 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 целых чисел. Необходимо определить количество элементов этого массива, для которых истинно логическое выражение  $(A[i] \neq 10) \rightarrow (3 \cdot A[i] \geq 9)$ . Например, в массиве из 4-х элементов, равных соответственно 3; -2; 10; 0, имеется два числа — 3 и 10, для которых вышеуказанное логическое выражение истинно.

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

Паскаль	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>
Бейсик	Python
<pre> <b>Const</b> N=2018 <b>Dim</b> A(N-1), I, M <b>As</b> Integer <b>For</b> I=0 <b>To</b> N-1   Input A(I) <b>Next</b> I ... <b>End</b> </pre>	<pre> # допускается также использование # целочисленной переменной m N=2018 a = [] <b>for</b> i <b>in</b> 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** Два игрока играют в игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами  $(x, y)$  в одну из трёх точек:

- в точку с координатами  $(x + 3, y)$ ;
- в точку с координатами  $(x, y + 2)$ ;
- в точку с координатами  $(x, y + 4)$ .

Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами  $(0, 0)$  больше 12 единиц. В начальный момент фишка находится в точке с **натуральными** координатами  $(x_0, y_0)$ , расстояние от которой до начала координат **меньше 12**.

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 количество начальных точек  $(x_0, y_0)$ , при которых первый игрок может выиграть в один ход. Обоснуйте, что найдены все нужные точки, и укажите выигрывающий ход для точки с минимальной суммой координат  $x_0 + y_0$ .  
б) Укажите такие начальные точки  $(x_0, y_0)$ , при которых первый игрок не может выиграть за один ход, но при любом его ходе второй игрок может выиграть своим первым ходом. Опишите выигрышную стратегию второго игрока.
- 2) Укажите хотя бы одну начальную точку  $(x_0, y_0)$ , при которой у первого игрока есть выигрышная стратегия, причём первый игрок не может выиграть за один ход и может выиграть своим вторым ходом независимо от того, как будет ходить его противник. Для указанной точки  $(x_0, y_0)$  опишите выигрышную стратегию первого игрока.
- 3) Укажите хотя бы одну начальную точку  $(x_0, y_0)$ , при которой:
  - у второго игрока есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре противника, и
  - у второго игрока нет стратегии, которая позволит ему гарантированно выиграть первым ходом.Для указанной точки  $(x_0, y_0)$  опишите выигрышную стратегию второго игрока. Постройте дерево всех партий, возможных при этой выигрышной стратегии второго игрока (в виде рисунка или таблицы).

**27** Дан набор из  $N$  неотрицательных целых чисел. Необходимо определить, какое количество единиц в двоичной записи этих чисел встречается чаще всего (если разное количество единиц встречается одинаково часто, укажите наибольшее из них). В ответе через пробел укажите найденное количество единиц и число из исходного набора, содержащее соответствующее количество единиц (если таких чисел несколько, то укажите наименьшее из них).

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  $N$  ( $1 \leq N \leq 10000$ ). В каждой из последующих  $N$  строк записано одно неотрицательное целое число, не превышающее 1000.

*Пример входных данных:*

5  
11  
7  
5  
9  
8

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

3 7

В приведённом наборе из 5 чисел имеется один элемент — 8, который имеет одну единицу в двоичной записи; два элемента, которые имеют две единицы в двоичной записи — 5, 9; два элемента — 11 и 7, которые имеют три единицы в двоичной записи. Чаще всего встречаются числа, содержащие в двоичной записи две или три единицы. Наибольшее количество единиц — 3, наименьшее из соответствующих чисел — 7.



## Ответы

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
5	64	3	5	971	6350	255	432
<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>
30	1	9	7	5	12	5	4
<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	
220	1	2286	21010	-2	392	324	

### 24 Содержание верного ответа:

- 1) При входных значениях  $n = 2$ ,  $x = 1$ ,  $a_2 = 2$ ,  $a_1 = 1$ ,  $a_0 = 0$  программа выведет число 4.
- 2) Программа выведет правильный ответ, равный 0, при входных значениях  $n = 0$ ,  $x = 0$ ,  $a_0 = 0$ . Возможны и другие варианты правильного ответа.
- 3) Программа содержит две ошибки: неверную инициализацию переменной  $p$  и неверное условие окончания цикла  $i \geq 1$ .

### Пример исправления на языке Паскаль

<b><i>Первая ошибка</i></b>	
Строка, содержащая ошибку	Исправленная строка
<code>p := x;</code>	<code>p := 0;</code>
<b><i>Вторая ошибка</i></b>	
Строка, содержащая ошибку	Исправленная строка
<code>for i := n downto 1 do begin</code>	<code>for i := n downto 0 do begin</code>

В программах на других языках ошибочные строки и их исправления аналогичны.

**25** Содержание верного ответа представлено на четырёх языках программирования:

Паскаль	C++
<pre> m:=0; for i:=0 to N-1 do   if (a[i]=10) or (3*a[i]&gt;=9) then     m:=m+1; writeln(m); </pre>	<pre> m=0; for (i=0; i&lt;N; i++)   if (a[i]==10    3*a[i]&gt;=9)     m++; std::cout&lt;&lt;m; </pre>
Бейсик	Python
<pre> M=0 ForI=0 To N-1   If A(i)=10 Or 3*A(i)&gt;=9 Then     M=M+1   End If Next I Print M </pre>	<pre> m=0 for i in range(N):   if a[i]==10 or 3*a[i]&gt;=9:     m+=1 print(m) </pre>

**Примечание.** Вместо условий  $(a[i]=10)$  **or**  $(3*a[i]>=9)$  может использоваться только условие  $3*a[i]>=9$ .

**26** Содержание верного ответа:

**Задание 1.** а) 49 точек. Для точки  $(1, 8)$  выигрывающий ход —  $(1, 8 + 4)$ . б) например  $(6, 6)$ .

**Задание 2.** Примеры точек:  $(3, 6)$ ,  $(6, 2)$  и  $(6, 4)$ . Первым ходом первый игрок передвинет фишку в точку  $(6, 6)$ .

**Задание 3.** Из начальной точки  $(3, 4)$  можно передвинуть фишку в  $(3, 8)$  (второй игрок выигрывает первым ходом) и в  $(3, 6)$  или  $(6, 4)$  (обе проигрышные и рассмотрены в предыдущем пункте).

**27** Содержание верного ответа представлено на четырёх языках программирования.

*Задание А. Программа неэффективна ни по времени, ни по памяти*

PascalABC.NET (version 3.4)	C++
<pre> <b>function</b> calc_ones(x : integer) : integer; <b>begin</b>   Result := 0;   <b>while</b> x &gt; 0 <b>do begin</b>     Result += x <b>mod</b> 2;     x := x <b>div</b> 2   <b>end</b> <b>end;</b>  <b>begin</b>   <b>var</b> N : integer;   <b>var</b> a : <b>array</b>[1..10000] <b>of</b> integer;   readln(N);   <b>for var</b> i := 1 <b>to</b> N <b>do</b>     readln(a[i]);   <b>var</b> max_cnt := 0; <b>var</b> cnt_ones := 0;   <b>var</b> min_num := 1001;   <b>for var</b> i := 1 <b>to</b> N <b>do begin</b>     <b>var</b> ones := calc_ones(a[i]);     <b>var</b> cnt := 0;     <b>for var</b> j := 1 <b>to</b> N <b>do</b>       <b>if</b> calc_ones(a[j]) = ones <b>then</b>         cnt += 1;     <b>if</b> cnt &gt; max_cnt <b>then begin</b>       max_cnt := cnt; cnt_ones := ones;       min_num := a[i]     <b>end else if</b> cnt = max_cnt <b>then</b>       <b>if</b> cnt_ones &lt; ones <b>then begin</b>         cnt_ones := ones; min_num := a[i]       <b>end else if</b> (cnt_ones = ones)         <b>and</b> (min_num &gt; a[i]) <b>then</b>         min_num := a[i]     <b>end;</b>     Print(cnt_ones, min_num)   <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>using</b> namespace std;  <b>int</b> calc_ones(<b>int</b> x) {   <b>int</b> cnt = 0;   <b>while</b> (x &gt; 0) {     cnt += x % 2; x /= 2;   }   <b>return</b> cnt; }  <b>int</b> main() {   <b>int</b> N, a[10000];   cin &gt;&gt; N;   <b>for</b> (<b>int</b> i = 0; i &lt; N; i++)     cin &gt;&gt; a[i];   <b>int</b> max_cnt = 0, cnt_ones = 0;   <b>int</b> min_num = 1001;   <b>for</b> (<b>int</b> i = 0; i &lt; N; i++) {     <b>int</b> ones = calc_ones(a[i]);     <b>int</b> cnt = 0;     <b>for</b> (<b>int</b> j = 0; j &lt; N; j++)       <b>if</b> (calc_ones(a[j]) == ones)         cnt++;     <b>if</b> (cnt &gt; max_cnt) {       max_cnt = cnt; cnt_ones = ones;       min_num = a[i];     } <b>else if</b> (cnt == max_cnt)       <b>if</b> (cnt_ones &lt; ones) {         cnt_ones = ones; min_num = a[i];       } <b>else if</b> (cnt_ones == ones &amp;&amp;         min_num &gt; a[i]) min_num = a[i];   }   cout &lt;&lt; cnt_ones &lt;&lt; " " &lt;&lt; min_num;   <b>return</b> 0; } </pre>

Visual Basic 2017	Python 3
<pre> <b>Module</b> Prog27 <b>Function</b> Calc_ones(X)     <b>Dim</b> Result=0     <b>While</b> X&gt;0         Result += X <b>Mod</b> 2 : X = X \ 2     <b>End While</b>     Calc_ones = Result <b>End Function</b>  <b>Sub</b> Main()     <b>Dim</b> N, A(10000), I, J <b>As</b> Integer     N=Console.ReadLine()     <b>For</b> I=0 <b>To</b> N-1         A(I)=Console.ReadLine()     <b>Next</b>     <b>Dim</b> max_cnt=0 : <b>Dim</b> cnt_ones=0     <b>Dim</b> min_num=1001     <b>For</b> I=0 <b>To</b> N-1         <b>Dim</b> ones=Calc_ones(A(I))         <b>Dim</b> cnt=0         <b>For</b> J=0 <b>To</b> N-1             <b>If</b> Calc_ones(A(J))=ones <b>Then</b>                 cnt +=1             <b>End If</b>         <b>Next</b>         <b>If</b> cnt&gt;max_cnt <b>Then</b>             max_cnt=cnt : cnt_ones=ones             min_num=A(I)         <b>ElseIf</b> cnt=max_cnt <b>Then</b>             <b>If</b> cnt_ones&lt;ones <b>Then</b>                 cnt_ones=ones : min_num=A(I)             <b>ElseIf</b> cnt_ones=ones <b>And</b> min_num&gt;a(i) <b>Then</b>                 min_num=A(I)             <b>End If</b>         <b>End If</b>     <b>Next</b>     Console.WriteLine(cnt_ones)     Console.WriteLine(min_num) <b>End Sub</b> <b>End Module</b> </pre>	<pre> <b>def</b> calc_ones(x) :     cnt=0     <b>while</b> x&gt;0 :         cnt+=x%2         x//=2     <b>return</b> cnt  N, a=<b>int</b>(input()), [] <b>for</b> i <b>in</b> range(N) :     a.append(<b>int</b>(input())) max_cnt, cnt_ones=0, 0 min_num=1001 <b>for</b> i <b>in</b> range(N) :     ones=calc_ones(a[i])     cnt=0     <b>for</b> j <b>in</b> range(N) :         <b>if</b> calc_ones(a[j])==ones :             cnt +=1         <b>if</b> cnt&gt;max_cnt:             max_cnt=cnt             cnt_ones=ones             min_num=a[i]         <b>elif</b> cnt==max_cnt :             <b>if</b> cnt_ones&lt;ones :                 cnt_ones=ones                 min_num=a[i]             <b>elif</b> (cnt_ones==ones <b>and</b> min_num&gt;a[i]) :                 min_num=a[i]     print(cnt_ones, min_num) </pre>

**Задание Б. Программа эффективна по времени и по памяти**

Паскаль	C++
<pre> <b>var</b> ones,     numbers: <b>array</b>[0..9] <b>of</b> integer;     N, i, x, y, cnt,     max_cnt, min_num: integer; <b>begin</b>     <b>for</b> i:=0 <b>to</b> 9 <b>do begin</b>         ones[i] := 0;         numbers[i] := 1001     <b>end;</b>     readln(N);     <b>for</b> i:=0 <b>to</b> N-1 <b>do begin</b>         readln(x);         y:=x;         count := 0;         <b>while</b> y&gt;0 <b>do begin</b>             cnt := cnt+y <b>mod</b> 2;             y:=y <b>div</b> 2         <b>end;</b>         ones[cnt] := ones[cnt]+1;         <b>if</b> numbers[cnt]&gt;x <b>then</b>             numbers[cnt] := x         <b>end;</b>         max_cnt := 0;         min_num := numbers[0];         <b>for</b> i:=1 <b>to</b> 9 <b>do</b>             <b>if</b> ones[max_cnt] &lt;= ones[i] <b>then</b>                 <b>begin</b>                     max_cnt := i;                     min_num := numbers[i]                 <b>end else</b>                     <b>if</b> (ones[max_cnt]=ones[i])                     <b>and</b> (min_num&gt;numbers[i]) <b>then</b>                         min_num := numbers[i];         write(max_cnt, ' ', min_num)     <b>end.</b> </pre>	<pre> #include &lt;iostream&gt; <b>using</b> namespace std; <b>int</b> main() {     <b>int</b> ones[10], numbers[10];     <b>for</b> (<b>int</b> i=0; i&lt;10; i++) {         ones[i]=0;         numbers[i]=1001;     }     <b>int</b> N, x;     cin&gt;&gt;N;     <b>for</b> (<b>int</b> i=0; i&lt;N; i++) {         cin&gt;&gt;x;         <b>int</b> y=x, cnt=0;         <b>while</b> (y&gt;0) {             cnt+=y%2;             y/=2;         }         ones[cnt]++;         <b>if</b> (numbers[cnt]&gt;x)             numbers[cnt]=x;     }     <b>int</b> max_cnt=0,         min_num=numbers[0];     <b>for</b> (<b>int</b> i=1; i&lt;10; i++)         <b>if</b> (ones[max_cnt]&lt;=ones[i]) {             max_cnt=i;             min_num=numbers[i];         } <b>else</b>             <b>if</b> (ones[max_cnt]==ones[i]                 &amp;&amp;min_num&gt;numbers[i])                 min_num=numbers[i];     cout&lt;&lt;max_cnt&lt;&lt;" "&lt;&lt;min_num;     <b>return</b> 0; } </pre>

Бейсик	Python
<pre>Dim Ones(10), Numbers(10) As Integer Dim N, I, X, Y, Cnt As Integer Dim Max_cnt, Min_num As Integer For I=0 To 9     Ones(i)=0 : Numbers(i)=1001 Next I Input(N) For I=0 To N-1     Input(X)     Y=X : Cnt=0     While Y&gt;0         Cnt=Cnt+Y Mod 2         Y=Y\2     End While     Ones(Cnt)=Ones(Cnt)+1     If Numbers(Cnt)&gt;X Then         Numbers(Cnt)=X     End If Next I Max_cnt=0 : Min_num=Numbers(0) For I=1 To 9     If Ones(Max_cnt)&lt;=Ones(I) Then         Max_cnt=I         Min_num=Numbers(I)     Else         If Ones(Max_cnt)=Ones(I) And Min_num&gt;Numbers(I) Then             Min_num=Numbers(I)         End If     End If Next I Print Max_cnt, Min_num</pre>	<pre>ones = [0] * 10 numbers = [1001] * 10 for i in range(int(input())):     x = int(input())     y, cnt = x, 0     while y &gt; 0:         cnt += y % 2         y //= 2     ones[cnt] += 1     if numbers[cnt] &gt; x:         numbers[cnt] = x max_cnt = 0 min_num = numbers[0] for i in range(1, 10):     if ones[max_cnt] &lt;= ones[i]:         max_cnt = i         min_num = numbers[i]     else:         if (ones[max_cnt] == ones[i] and min_num &gt; numbers[i]):             min_num = numbers[i] print(max_cnt, min_num)</pre>