

## Информатика, 9 класс, решения

### Вариант 1.

№	Правильный ответ	Балл	Прим
1.	<p>Если перевести все числа в двоичный вид, при этом записав длинную последовательность нулей как <math>\{0\}_N</math>, мы получим следующее выражение <math>1\{0\}_{2N-1}1*1\{0\}_{3N-2}11 = 1\{0\}_{3N-2}11\{0\}_{2N+1}\{0\}_{3N-2}11=1\{0\}_{2N-1}1\{0\}_{N-2}11\{0\}11_{2N}</math>. В этой записи ровно <math>5N-3</math> нуля. Решаем неравенство <math>5N-3 &gt; 900000</math>, получаем 180001</p>	16	
2.	<p>1) Выигрывает второй игрок, у первого игрока есть возможность разрезать только на прямоугольники 1 на 3 и 2 на 3. После этого второй игрок разрезает 2 на 3 на две одинаковые части. Больше ходов сделать нельзя, так как разрезать прямоугольник 1 на 3, не нарушив правило.</p> <p>2) Выигрывает первый игрок, разрезав на две одинаковые части по 3 на 11. Дальше надо действовать симметрично, если второй игрок сделает с одним куском что-то, первый игрок делает то же самое, но со вторым.</p> <p>3) В этом варианте игры все прямоугольники будут размера 1 на n, потому для удобства будем просто писать в таком случае кусочек размера n. Дальше заметим, что куски размером 2 и 3 нельзя разрезать, а 4 и 5 разрезать можно разрезать только один раз. Потому если кусочек размером 2 и 3 появляется, его можно выбросить, на игру он не влияет. Первым ходом надо разрезать прямоугольник на куски размером 9 и 4. Возможные ответы второго игрока: (9, 2, 2), (2, 7, 4), (3, 6, 4), (4, 5, 4). Выбросим ненужные куски. В первом случае ходим (4, 5), остаётся чётное число ходов и второй игрок проигрывает. Во всех остальных случаях можно сделать также, оставив второму игроку два куска размером 4 или 5 и несколько неразрезаемых кусочков. Второй игрок разрезает любой из них, затем первый игрок разрезает и ходы заканчиваются, первый игрок выигрывает.</p> <p>4) Кусок с чётной стороной можно, как в случае 2, разрезать пополам и повторять действия противника. Кусок со сторонами 1 и 11 же является проигрышным, и если Полина сделает в нём ход, то Вера тоже сможет сделать в нём ход, ведущий к проигрышу Полины. Рассмотрим все случаи, Полина может разрезать этот кусок на (9,2), (8,3), (7,4), (6,5). В первом случае 9 надо разрезать на 4 и 5, тогда остаётся два хода в</p>	21	

	<p>независимости от игры. Во втором 8 разрежем на 4 и 4. В двух оставшихся можно от 7 или 6 отрезать так, чтобы получился кусочек размером 4, и тоже останется ровно 2 хода.</p>		
3.	<p>Ответ: 32. Для ответа надо построить таблицу истинности для двух выражений в начале импликаций и увидеть, что есть только пять строк с двумя нулями. Если хотя бы же одна из этих функций равна 1, то и F обязана равняться 1. Для этих же пяти строк значение функции может быть любым, итого мы получаем 2 в пятой различных функций.</p>	18	
4.	<p>Вместо того, чтобы программировать эту функцию и запускать её напрямую, лучше понять, что она делает. Она выдаёт число вида <math>5*u + 1*v</math>, где <math>v</math> это число двоек, которые делят число. <math>U</math> же это количество 4, которые мы можем отнять от числа, пока оно не станет меньшим или равным 1. Например, для 3 и 5 это 1, для 7 и 9 это 2. Его можно вычислить по формуле <math>(n+1)//4</math>, где два слэша это целочисленное деление с округлением вниз. Итого если мы хотим получить 5, нам нужно минимальное число 3, если 10, то 7, если 15, то 11. То есть мы делим изначально заданное <math>n</math> на 5, умножаем на 4 и отнимаем 1. Чтобы получить числа, не делящиеся на 5, нам нужно ещё умножить его на два в зависимости от остатка. Итого надо просто в программе реализовать следующее выражение <math>(n//5*4 - 1)*2^{(n\%5)}</math>.</p> <pre> n = int(input()) B, ost = n // 5, n % 5 if B &gt; 0:     m1 = B * 4 - 1 else:     m1 = 1 m2 = 2 ** ost print(m1 * m2) </pre> <p>Ответы на тесты: 174,13040, 6550, 615856</p>	20	
5.	<pre> from functools import lru_cache @lru_cache() def f(n):     if n == 0:         return 1     if n &lt;= 3:         return 2 * n     return f(n-1) + f(n-3) + f(n-4) print(f(n) - n - 1) </pre> <p>Можно сделать без кэширования, создав массив, который</p>	25	

будет постепенно заполняться. Ответы на тесты: 2902, 6676557, 62423799720		
--	--	--

Вариант 2.

№	Правильный ответ	Балл	Прим
1.	Если перевести все числа в двоичный вид, при этом записав длинную последовательность нулей как $\{0\}_N$ , мы получим следующее выражение $1\{0\}_{2N-1}1*1\{0\}_{3N-2}11 = 1\{0\}_{3N-2}11\{0\}_{2N+1}\{0\}_{3N-2}11 = 1\{0\}_{2N-1}1\{0\}_{N-2}11\{0\}11_{2N}$ . В этой записи ровно $5N-5$ нуля. Решаем неравенство $5N-5 > 800000$ , получаем 160002	16	
2.	<p>1) Выигрывает второй игрок. Если разрез вертикальный и отрезается кусочек 1 на 3 или 2 на 3, то надо дальше сделать вертикальный разрез так, чтобы остался кусочек размера 3 на 3 (разрезать его можно только одним способом, после которого останется Вере лишь доразрезать его). Если же Полина делает горизонтальный разрез, то Вера делает вертикальный разрез, отрезая кусочек 2 на 1 сбоку. Если Полина продолжить делать вертикальный разрез, то их будет ровно 3 плюс разрез, который можно сделать на кусочке 1 на 5. Если горизонтальный, то останется нечётное количество разрезов Вере, и она выигрывает.</p> <p>2) Выигрывает первый игрок, разрезав на две одинаковые части по 4 на 11. Дальше надо действовать симметрично, если второй игрок сделает с одним куском что-то, первый игрок делает то же самое, но со вторым.</p> <p>3) В этом варианте игры все прямоугольники будут размера 1 на n, потому для удобства будем просто писать в таком случае кусочек размера n. Дальше заметим, что куски размером 2 и 3 нельзя разрезать, а 4 и 5 разрезать можно разрезать только один раз. Кусок размера 6 можно разрезать 1 или 2 раза, а кусок размера 7 ровно 2 раза и никак иначе. Потому если кусочек размером 2 и 3 появляется, его можно выбросить, на игру он не влияет. Первым ходом можно сделать следующий варианты: (9), (8), (7, 4), (6, 5). Любой из них можно свести к (4, 5) или (4, 4). Тогда остаётся всего лишь два разреза, и второй игрок выигрывает, потому что именно он делает последний разрез.</p>	21	

	<p>4) Кусок с чётной стороной можно, как в случае 2, разрезать пополам и повторять действия противника. Кусок со сторонами 1 и 11 же является проигрышным, и если Полина сделает в нём ход, то Вера тоже сможет сделать в нём ход, ведущий к проигрышу Полины. Рассмотрим все случаи, Полина может разрезать этот кусок на (9,2), (8,3), (7,4), (6,5). В первом случае 9 надо разрезать на 4 и 5, тогда остаётся два хода в независимости от игры. Во втором 8 разрежем на 4 и 4. В двух оставшихся можно от 7 или 6 отрезать так, чтобы получился кусочек размером 4, и тоже останется ровно 2 хода.</p>		
3.	<p>16. Для ответа надо построить таблицу истинности для двух выражений в начале импликаций и увидеть, что есть только 4 строки с двумя нулями. Если хотя бы же одна из этих функций равна 1, то и F обязана равняться 1. Для этих же 4 строк значение функции может быть любым, итого мы получаем 2 в четвёртой различных функций.</p>	18	
4.	<p>Вместо того, чтобы программировать эту функцию и запускать её напрямую, лучше понять, что она делает. Она выдаёт число вида <math>3*u + 1*v</math>, где <math>v</math> это число двоек, которые делят число. <math>U</math> же это количество 4, которые мы можем отнять от числа, пока оно не станет меньшим или равным 1. Например, для 3 и 5 это 1, для 7 и 9 это 2. Его можно вычислить по формуле <math>(n+1)//4</math>, где два слэша это целочисленное деление с округлением вниз. Итого если мы хотим получить 3, нам нужно минимальное число 3, если 6, то 7, если 9, то 11. То есть мы делим изначально заданное <math>n</math> на 3, умножаем на 4 и отнимаем 1. Чтобы получить числа, не делящиеся на 3, нам нужно ещё умножить его на два в зависимости от остатка. Итого надо просто в программе реализовать следующее выражение <math>(n//3*4 - 1)*2^{(n\%3)}</math>.</p> <pre> n = int(input()) B, ost = n // 3, n % 3 if B &gt; 0:     m1 = B * 4 - 1 else:     m1 = 1 m2 = 2 ** ost print(m1 * m2) </pre> <p>Ответы на тесты: 147, 2726, 10918, 256620</p>	20	
5.	<pre> from functools import lru_cache @lru_cache() </pre>	25	

<pre>def f(n):     if n == 0:         return 1     if n &lt;= 3:         return 2 * n     return f(n-1) + f(n-3) + f(n-4) print(f(n) - n - 1)</pre> <p>Можно сделать без кэширования, создав массив, который будет постепенно заполняться.</p> <p>Ответы на тесты: 2902, 6676557, 62423799720</p>		
---	--	--

### Вариант 3.

№	Правильный ответ	Балл	Прим
1.	Если перевести все числа в двоичный вид, при этом записав длинную последовательность нулей как $\{0\}_N$ , мы получим следующее выражение $1\{0\}_{2N-1}1*1\{0\}_{3N-2}11 = 1\{0\}_{3N-2}11\{0\}_{2N+1}\{0\}_{3N-2}11=1\{0\}_{2N-1}1\{0\}_{N-2}11\{0\}_{2N-2}11$ . В этой записи $2N-1$ самым первым будет равно 799. Это возможно при $N = 400$ .	16	
2.	<p>1) Выигрывает второй игрок, у первого игрока есть возможность разрезать только на прямоугольники 1 на 3 и 2 на 3. После этого второй игрок разрезает 2 на 3 на две одинаковые части. Больше ходов сделать нельзя, так как разрезать прямоугольник 1 на 3, не нарушив правило.</p> <p>2) Выигрывает первый игрок, разрезав на две одинаковые части по 7 на 7. Дальше надо действовать симметрично, если второй игрок сделает с одним куском что-то, первый игрок делает то же самое, но со вторым.</p> <p>3) В этом варианте игры все прямоугольники будут размера 1 на n, потому для удобства будем просто писать в таком случае кусочек размера n. Дальше заметим, что куски размером 2 и 3 нельзя разрезать, а 4 и 5 разрезать можно разрезать только один раз. Потому если кусочек размером 2 и 3 появляется, его можно выбросить, на игру он не влияет. Первым ходом надо разрезать прямоугольник на куски размером 9 и 4. Возможные ответы второго игрока: (9, 2, 2), (2, 7, 4), (3, 6, 4), (4, 5, 4). Выбросим ненужные куски. В первом случае ходим (4, 5), остаётся чётное число ходов и второй игрок проигрывает. Во всех остальных случаях можно сделать также, оставив второму игроку два куска размером 4 или 5 и несколько неразрезаемых кусочков. Второй игрок разрезает любой из них, затем первый</p>	21	

	<p>игрок разрезает и ходы заканчиваются, первый игрок выигрывает.</p> <p>4) Кусок с чётной стороной можно, как в случае 2, разрезать пополам и повторять действия противника. Кусок со сторонами 1 и 11 же является проигрышным, и если Полина сделает в нём ход, то Вера тоже сможет сделать в нём ход, ведущий к проигрышу Полины. Рассмотрим все случаи, Полина может разрезать этот кусок на (9,2), (8,3), (7,4), (6,5). В первом случае 9 надо разрезать на 4 и 5, тогда остаётся два хода в независимости от игры. Во втором 8 разрежем на 4 и 4. В двух оставшихся можно от 7 или 6 отрезать так, чтобы получился кусочек размером 4, и тоже останется ровно 2 хода.</p>		
3.	<p>32. Для ответа надо построить таблицу истинности для двух выражений в начале импликаций и увидеть, что есть только пять строк с двумя нулями. Если хотя бы же одна из этих функций равна 1, то и F обязана равняться 1. Для этих же пяти строк значение функции может быть любым, итого мы получаем 2 в пятой различных функций.</p>	18	
4.	<p>Вместо того, чтобы программировать эту функцию и запускать её напрямую, лучше понять, что она делает. Она выдаёт число вида <math>3*u + 1*v</math>, где <math>v</math> это число двоек, которые делят число. <math>U</math> же это количество 6, которые мы можем отнять от числа, пока оно не станет меньшим или равным 1. Например, для 3, 5 и 7 это 1, для 9, 11 и 13 это 2. Его можно вычислить по формуле <math>(n+3)//6</math>, где два слэша это целочисленное деление с округлением вниз. Итого если мы хотим получить 3, нам нужно минимальное число 3, если 6, то 9, если 9, то 15. То есть мы делим изначально заданное <math>n</math> на 3, умножаем на 6 и отнимаем 3. Чтобы получить числа, не делящиеся на 3, нам нужно ещё умножить его на два в зависимости от остатка. Итого надо просто в программе реализовать следующее выражение <math>(n//3*6 - 3)*2^{(n\%3)}</math>.</p> <pre> n = int(input()) B, ost = n // 3, n % 3 if B &gt; 0:     m1 = B * 6 - 3 else:     m1 = 1 m2 = 2 ** ost print(m1 * m2) </pre> <p>Ответы на тесты: 219, 4086, 16374, 384924</p>	20	

5.	<pre> from functools import lru_cache @lru_cache() def f(n):     if n == 0:         return 1     if n &lt;= 3:         return 2 * n     return f(n-1) + f(n-3) + f(n-4) print(f(n) - n - 1) </pre> <p>Можно сделать без кэширования, создав массив, который будет постепенно заполняться.          Ответы на тесты: 2902, 6676557, 62423799720</p>	25	
----	--	----	--

Вариант 4

№	Правильный ответ	Балл	Прим
1.	<p>Если перевести все числа в двоичный вид, при этом записав длинную последовательность нулей как <math>\{0\}_N</math>, мы получим следующее выражение <math>\{1\}_{2N} * 1\{0\}_{N-1} = \{1\}_{2N}\{0\}_N + \{1\}_{2N} = 1\{0\}_N\{1\}_{N-1}0\{1\}_N</math>. В этой записи <math>2N</math> единиц, и оно больше 600 при <math>N = 301</math>.</p>	16	
2.	<p>1) Выигрывает второй игрок, у первого игрока есть возможность разрезать только на прямоугольники 1 на 3 и 2 на 3. После этого второй игрок разрезает 2 на 3 на две одинаковые части. Больше ходов сделать нельзя, так как нельзя разрезать прямоугольник 1 на 3, не нарушив правило.</p> <p>2) Выигрывает первый игрок, разрезав на две одинаковые части по 7 на 7. Дальше надо действовать симметрично, если второй игрок сделает с одним куском что-то, первый игрок делает то же самое, но со вторым.</p> <p>3) В этом варианте игры все прямоугольники будут размера 1 на n, потому для удобства будем просто писать в таком случае кусочек размера n. Дальше заметим, что куски размером 2 и 3 нельзя разрезать, а 4 и 5 разрезать можно разрезать только один раз. Кусок размера 6 можно разрезать 1 или 2 раза, а кусок размера 7 ровно 2 раза и никак иначе. Потому если кусочек размером 2 и 3 появляется, его можно выбросить, на игру он не влияет. Первый игрок должен разрезать так (5, 10). Дальше возможны следующие ходы: (10), (5, 8), (5, 7), (5, 4, 6), (5, 5, 5). Любую из этих ситуаций можно свести к (5, 5) или (5, 4). Тогда остаётся всего лишь два разреза, и второй игрок выигрывает, потому что именно он делает последний разрез.</p>	21	

	<p>4) Кусок с чётной стороной можно, как в случае 2, разрезать пополам и повторять действия противника. Кусок со сторонами 1 и 11 же является проигрышным, и если Полина сделает в нём ход, то Вера тоже сможет сделать в нём ход, ведущий к проигрышу Полины. Рассмотрим все случаи, Полина может разрезать этот кусок на (9,2), (8,3), (7,4), (6,5). В первом случае 9 надо разрезать на 4 и 5, тогда остаётся два хода в независимости от игры. Во втором 8 разрежем на 4 и 4. В двух оставшихся можно от 7 или 6 отрезать так, чтобы получился кусочек размером 4, и тоже останется ровно 2 хода.</p>		
3.	<p>0</p> <p>Для ответа надо построить таблицу истинности для двух выражений в начале первой и конце второй импликаций и увидеть, что есть в ней строка, где первая функция равна 1, а вторая нулю. Тогда F должна равняться одновременно 0 и 1, что невозможно.</p>	18	
4.	<p>Вместо того, чтобы программировать эту функцию и запускать её напрямую, лучше понять, что она делает. Она выдаёт число вида <math>5*u + 1*v</math>, где <math>v</math> это число троек, которые делят число. <math>U</math> же это количество 3, которые мы можем отнять от числа, пока оно не станет меньшим или равным 1. Например, для 2 и 4 это 1, для 5 и 7 это 2. Итого если мы хотим получить 5, нам нужно минимальное число 2, если 10, то 5, если 15, то 8. То есть мы делим изначально заданное <math>n</math> на 5, умножаем на 3 и отнимаем 1. Чтобы получить числа, не делящиеся на 5, нам нужно ещё умножить его на два в зависимости от остатка. Итого надо просто в программе реализовать следующее выражение <math>(n//5*3 - 1)*3^{(n\%5)}</math>.</p> <pre>n = int(input()) B, ost = n // 5, n % 5 if B &gt; 0:     m1 = B * 3 - 1 else:     m1 = 1 m2 = 3 ** ost print(m1 * m2)</pre> <p>Ответы на тесты: 195, 49491, 7368, 2338308</p>	20	
5.	<pre>from functools import lru_cache @lru_cache() def f(n):     if n == 0:         return 1</pre>	25	



	<pre>if n &lt;= 3:     return 2 * n return f(n-1) + f(n-3) + f(n-4) print(f(n) - 3 - (n-1)*n//2)</pre> <p>Можно сделать без кэширования, создав массив, который будет постепенно заполняться.</p> <p>Ответы на тесты: 3008, 6677023, 62423800946</p>		
--	--	--	--

### Информатика, 9 класс, критерии

1. Приведено полное обоснованное решение - 16 баллов  
Получено верное неравенство, но при его решении допущены ошибки - 14 баллов  
Выписано, как выглядят степени 2 в двоичной системе, при этом дальнейшие продвижения отсутствуют - 2-4 балла.
2. Рассмотрены все 4 случая, приведено полное описание выигрышной стратегии - 21 балл  
Рассмотрены и подробно описаны только 3 случая выигрышной стратегии – 16 баллов  
Рассмотрены и подробно описаны только 2 случая выигрышной стратегии – 11 баллов  
Рассмотрены и подробно описаны только 1 случай выигрышной стратегии – 5 баллов  
За неполное описание выигрышной стратегии баллы пропорционально снижаются.
3. Верно построены таблицы истинности. Приведено полное обоснованное решение – 18 баллов  
Верно построены таблицы истинности. Описано, какие строки таблицы истинности являются определяющими, но при этом сделаны неправильные выводы – 14 баллов  
Верно построены таблицы истинности для двух выражений, при этом дальнейшие продвижения отсутствуют - 4 балла  
Таблицы истинности выражений построены, но содержат незначительные ошибки.  
Дальнейшие продвижения отсутствуют – 2 балла
4. На все тесты программа выдает верный ответ – 20 баллов.  
Программа выдает верный ответ на 3 теста из 4 – 15 баллов.  
Программа выдает верный ответ на 2 теста из 4 – 10 баллов.  
Программа выдает верный ответ на 1 тест из 4 – 5 баллов.
5. На все тесты программа выдала верный ответ за время экзамена – 25 баллов.  
Программа выдает верный ответ на 2 теста из 3 – 16 баллов.  
Программа выдает верный ответ на 1 тест из 3 – 8 баллов.