

Информатика.10 класс

Решения и ответы

Вариант 1.

№	Правильный ответ
1.	Нам неизвестно, добавила ли программа двойку к числу или нет, поэтому проверим оба варианта. Переведём число 11111110 в шестеричную систему счисления. Это число 254 в десятичной, в шестеричной выглядит как 1102, однако нельзя получить такой ответ (ведь должно было получиться число в двоичной записи). Следовательно, программа добавила двойку. Отнимем её. Получим 11111100 в двоичной (число единичек в записи чётно) и 1100 в шестеричной. Очевидно, что программа не могла получить число 1100 в двоичной записи без добавления двойки, так как если бы она получила число 1100 после первого шага, она бы добавила к нему двойку, следовательно, отнимем двойку и переведём в шестеричную, получив число 14. Ответ 14.
2.	Очевидно, что если в одной из куч камней n, а в другой ноль, или же в обеих кучах камней ровно n, то игрок, которому досталась такая ситуация, сразу выигрывает. Значит, игрок проигрывает, когда в его ситуации он может перейти только к такой выигрышной для противника ситуации. Например, при числе камней 2 и 1 любой ход приводит к одной из ситуаций из начала. Значит, приходиться к такой позиции нельзя. Отойдём чуть повыше и переберём маленькие значения камней. Окажется, что ситуации 3 и 5, а также 4 и 7 схожи с 1 и 2; несложно убедиться перебором, что при любом ходе соперник сможет его снова поставить на заведомо проигрышную позицию. В случае числа камней 3 и 102 выигрывает первый игрок, он забирает 97 камней из второй кучи и оставляет позицию 3 и 5. Как бы не ходил второй, можно будет свести к позиции 1 и 2, которая заведомо проигрышная. В случае числа камней 4 и 7 выигрывает второй игрок, потому что после любого хода первого игрока окажется либо одна куча с 3 камнями или меньше (тогда можно будет свести позицию к 3 и 5 или 2 и 1), либо разница между кучами станет 2 или меньше, и тогда тоже можно будет свести позицию к 3 и 5 или 2 и 1.
3.	Можно заметить, что запрос $Рост < 150$ и $Вес < 50$ является подмножеством первого запроса, при этом является всем тем, что входит в первый запрос, но не входит во 2 и 3. При этом 4 запрос – это пересечение 2 и 3. Значит, в объединение 2 и 3 запроса входят $90 + 107 - 31 = 166$ запросов. Теперь от числа записей в первом отнимем полученное число запросов, итого 35 запросов. Для второго же запроса

	возможное число записей начинается от 35, так как он содержит все записи из предыдущего, и заканчивается 94, так как теоретически туда могут входить все записи, не вошедшие в 3 запрос.
4.	Маски можно подобрать самые разные, например <code>??x??.*у*</code>. Действительно, эта маска даёт общие имена файлов в связке с первой и второй, а именно <code>xzxwu.xu</code> и <code>wxxuw.xu</code>, но нет ни одного имени, которое бы удовлетворяло сразу трём маскам, так как в этом имени до точки должно быть ровно 5 букв, третья <code>x</code>, при этом из первой маски у нас выходит условие <code>?z?w?</code>, значит, в середине будет конструкция <code>zxw</code>, но во второй маске до точки обязана быть <code>uw</code>, но её уже никак не добавить в <code>?zxw?</code>.
5.	Один из примеров кода на языке Python 3.7 <pre> N = int(input()) a = [0 for x in range(N+5)] a[1] = 1 a[2] = 1 a[3] = 2 for i in range(4, N+1): a[i] = a[i-1] + a[i-3] + 1 print(a[N]) </pre> Будем вычислять это рекурсивно. Либо после 1 мы сразу можем поставить 2, тогда у нас вариантов останется столько же, сколько и для последовательности на 1 меньшую, либо 1324, тогда мы окажемся с последовательностью на три меньшую, либо можно сразу идти до конца 1357..., а потом обратно одним единственным способом.

Вариант 2.

№	Правильный ответ
1.	Нам неизвестно, отняла ли программа двойку от числа или нет, поэтому проверим оба варианта. Переведём число 10000000 в пятеричную систему счисления. Это число 128 в десятичной, в пятеричной выглядит как 1003, однако программа не могла дать такой ответ. Следовательно, программа отняла двойку. Добавим её. Получим 10000010 в двоичной (число единичек в записи чётно) и 1010 в пятеричной. Очевидно, что программа не могла дать число 1010 в двоичной записи без отнимания двойки, так как если бы она получила число 1010 после первого шага, она бы отняла от него двойку, следовательно, добавим двойку и переведём в пятеричную, получив число 22. Ответ 22
2.	Очевидно, что если в одной из куч камней n, а в другой ноль, или же в обеих кучах камней ровно n, то игрок, которому досталась такая ситуация, сразу выигрывает. Значит, игрок проигрывает, когда в его ситуации он может перейти только к такой выигрышной для противника ситуации. Например, при числе камней 2 и 1 любой ход приводит к одной из ситуаций из начала. Значит, приходиться к такой позиции нельзя. Отойдём чуть повыше и переберём маленькие значения камней. Окажется, что ситуации 3 и 5, а также 4 и 7 схожи с 1 и 2; несложно убедиться перебором, что при любом ходе соперник сможет его снова поставить на заведомо проигрышную позицию. В случае числа камней 3 и 5 выигрывает второй игрок, потому что после любого хода первого игрока окажется либо одна куча с 2 камнями или меньше (тогда можно будет свести позицию к 2 и 1), либо разница между кучами станет 1, и тогда тоже можно будет свести позицию к 2 и 1. В случае же 6 и 3 камня выигрывает первый игрок, он сводит всё к ситуации 4 и 7 первым ходом. Как бы второй не ходил, первый игрок всегда может оставить его с числом камней 3 и 5 или 2 и 1, которые, как мы знаем, проигрышные для того, кто с них ходит.
3.	Можно заметить, что запрос <i>Рост <150 и Вес <50</i> является подмножеством первого запроса, при этом является всем тем, что входит в первый запрос, но не входит во 2 и 3. При этом 4 запрос – это пересечение 2 и 3. Значит, в объединение 2 и 3 запроса входят $120 + 109 - 50 = 179$ запросов. Теперь от числа записей в первом отнимем полученное число запросов, итого 54 запроса. Для второго же запроса возможное число записей начинается от 54, так как он содержит все записи из предыдущего, и заканчивается 113, так как теоретически туда могут входить все записи, не вошедшие во 2 запрос.
4.	Маски можно подобрать самые разные, например <code>???x?.*x*</code>. Действительно, эта маска даёт общие имена файлов в связке с первой и второй, а именно <code>xууzxx.ux</code> и <code>xwwwuxx.xz</code>, но нет ни одного имени, которое бы удовлетворяло сразу трём маскам, так как в этом имени до

	точки должно быть ровно 6 букв, вторая w и четвёртая u (из второй маски), первая x (из 1), пятая x, то есть xw?ux?, но при этом из условия первой маски обязана содержать конструкцию uz, которую уже никак не добавить в получившееся слово.
5.	Один из примеров кода на языке Python 3.7 <pre> N = int(input()) a = [0 for x in range(N+5)] a[1] = 1 a[2] = 1 a[3] = 2 for i in range(4, N+1): a[i] = a[i-1] + a[i-3] + 1 print(a[N]) </pre> Будем вычислять это рекурсивно. Либо после 1 Бельчонок сразу прыгает на 2, тогда у нас вариантов останется столько же, сколько и для лестницы на 1 ступеньку меньшую, либо с первой на 3, потом на 2 и на 4, тогда мы окажемся с лестницей на три ступеньки меньшей, либо можно сразу идти до конца 1357..., а потом обратно одним единственным способом.

Вариант 3.

№	Правильный ответ
1.	Рассмотрим остатки при делении числа 31 на 11 и 13, это 9 и 5. Один из этих остатков удвоили. Остаток 9 по модулю 11 всё равно, что 20, а 5 по модулю 13 всё равно, что 18. Тогда если первый остаток удвоили, изначально он был 10, то есть, по сути, добавили число 10. Тогда изначально было число 21, ну и действительно, наибольший остаток из двух возможных это 10. Рассуждая также во втором случае, получим число 22, у него наибольший остаток это 9 по модулю 13, прибавляя его, получаем 31. К каждому из этих чисел нужно ещё раз применить этот трюк, но можно заметить, что число 22 нельзя было получить, прибавляя остаток от деления на 11, только на 13, и оно могло быть получено только из числа 11, число 21 могло быть получено из 16 и больше никакого. Итого, два числа могло быть в начале: 16 и 11.
2.	Через поле, на котором стоит мышка, проведем какую-нибудь диагональ и поставим кошек на ее конце. На каждый ход мышки кошки ходят так, чтобы все три фигуры снова оказались на одной диагонали, а расстояние между кошками сократилось на одно поле (по диагонали). Такая стратегия позволяет кошкам в скором времени съесть мышку.
3.	Мы не знаем, сколько людей ходят на три кружка одновременно, но из последнего утверждения становится ясно, что их не может быть 0, иначе мы не могли бы посчитать средний рост, но и не 2, иначе тогда это множество людей совпало с тем множеством, которое ходит только на шахматы и футбол. Значит, такой человек ровно один. Теперь мы должны сложить вместе всех тех, кто ходит хотя бы на один конкретный кружок ($20+24+30$). Получим 74, но здесь мы посчитали дважды тех, кто ходит сразу на два кружка, и трижды того одного, который ходит на всё. Отнимем от $74-9-9-2 = 54$, теперь мы исправили предыдущее недоразумение, но совсем не посчитали единственного трёхкружковца, добавим его, получим 55.
4.	Минимум в имени будет содержаться 6 символов. Рассмотрим случаи, когда символов 6, 7 и 8. В случае шести символов маска становится такой $xuz.w?$, и у нас есть 4 возможности заменить вопрос на что-то. При 7 символах могут быть $x?uz.w?$ и $xuz.w?$, каждая из этих масок даёт $4*4 = 16$ вариантов, и наконец при 8 символах возможно 3 варианта: $x??uz.w?$, $x?uz?.w?$, $xuz??.w?$, каждая из масок даёт $4*4*4 = 64$ варианта, но первый и последний вариант имеют в пересечении 4 общих имени $xuzyz.w?$, их нужно будет при финальном подсчёте отнять. Итого $4 + 16 + 16 + 64 + 64 + 64 - 4 = 224$
5.	<pre> f = input().split() K = int(f[0]) N = int(f[1]) a = [0 for x in range(N+1)] a[0] = 1 </pre>

<pre>for i in range(1, N+1): d = 0 for j in range(K): if i-j-1 >= 0: d = d + a[i-j-1] a[i] = d print(a[N])</pre>
