

Открытый online турнир по программированию «Этот загадочный Python» Юго-Восточного образовательного округа

Центр цифрового образования «ИТ-куб» на базе КОГОАУ
«Вятский многопрофильный лицей» г. Вятские Поляны
12 мая 2026 года

Разбор задач

Фортран и медали

Если n кратно 4, то ответом будет 0.

Если n кратно 1, то нужно добавить, как минимум 3 медали, поэтому ищем $\min(3*a, b+a, c)$.

Если n кратно 2, то нужно добавить, как минимум 2 медали, поэтому ищем $\min(2*a, b, 2*c)$.

Если n кратно 3, то нужно добавить, как минимум 1 медаль, поэтому ищем $\min(a, b+c, 3*c)$.

Выводим найденное значение.

Кубатура шара

Проходимся по списку, и, если размер очередной размер заготовки (болванки) нас устраивает, помещаем его в плавильный аппарат. Сразу же проверяем, не переполнился ли контейнер. Если да, то добавляем единицу к счетчику, чистим контейнер.

Счетчик и будет результатом.

Тараканьи бега

Учитывая, что в каждой клетке могут находиться тараканы, можно организовать перебор, находя расстояние от каждой клетки до клетки с координатами (r, c) .

Но этого делать не следует. Очевидно, что минимальным временем будет то, когда таракан, находящийся в самой дальней клетке доберется до заданной клетки (r, c) .

Достаточно определить эти возможные расстояния. Так как движение будет только по горизонталям и вертикалям, результатом будет сумма максимальных расстояний $\max(r - 1, n - r) + \max(c - 1, m - r)$.

Бальные страсти

Вариантов решения множество. Один из возможных: создать вспомогательный массив из трех элементов, в который при считывании каждой тройки чисел a , b , c будем добавлять на соответствующие места 2, 1 или 0. Результат – итоговое содержимое трех ячеек этого массива.

И снова тараканы

Если количество тараканов равно двум, то Анфиса вынуждена обязательно сделать ход, результатом всегда будет цифра на последнем таракане.

В противном случае, Анфиса будет стараться играть оптимально, результатом на последнем таракане будет самая минимальная цифра из всех имеющихся.

Хитрая Анфиса

В задаче из имеющихся четных чисел только одно может дать результат. Если мы постоянно делим число на 2, то в результате мы получим 1. Остальные четные числа при добавлении к s не смогут превратить число в нечетное.

Но это смогут сделать нечетные числа.

Поэтому считаем количество нечетных k_1 и количество четных k_2 чисел. Если есть хотя бы одно четное число. То результатом будет $k_1 + 1$.

А что делать, если четных чисел нет. Тогда результатом будет $k_1 - 1$.

Удивительный водосток

Обозначим сумму трех чисел $a + b + c$ как x . Чтобы вписаться в отведенное время, нужно определить возможное количество циклов из трех минут, поделив n капель на сумму трех чисел, $p = n // x$.

Текущее количество минут будет равно $k = p * 3$.

Остальные возможные минуты проверяем вручную. Обозначим $n \% x$ за y . Если $y = 0$, то результатом будет k . Иначе, если $y \leq a$, то к результату добавим 1, иначе если $y \leq a + b$, то к результату добавим 2, иначе если $y \leq a + b + c$, то добавим 3.

Хитрый букмекер

Поиск каждого значения необходимо осуществлять справа. На каждом шаге находим \max ставку среди всех жокеев, которые расположены правее. Она считается по формуле $\max - a[i] + 1$.

После нахождения очередного значения нужно переопределять \max , включая текущее, уже просмотренное. Кроме того, нужно сравнивать разницу с 0, чтобы не уйти в минус.

Мудрый попугай

Можно создать для каждой из строк словарь, в котором будет подсчитано количество вхождений в строку каждой из букв.

Проходим по первой строке, проверяем, есть ли такая буква во втором словаре. Если есть, увеличиваем первый счетчик, и удаляем по единице из ячеек, соответствующим букве, в каждом из словарей.

Первый счетчик готов.

Вторично проходим по первой строке, проверяем в первом словаре, осталась ли еще данная буква. Затем обращаемся во второй словарь, ищем альтернативу букве (можно использовать коды букв). Если альтернатива букве есть, учитываем ее во втором счетчике и удаляем по единичке в найденных ячейках обоих словарей.

Выводим счетчики.

Субботник со скворцами

Если нет ни одной единицы, то результат равен 0.

Количество способов распределить липы на участке на 1 больше, чем количество рядом стоящих нулей, то есть, это же равно разности между соседними индексами единиц.

Для решения достаточно сохранить в отдельном списке позиции единиц и найти все разности между соседними элементами (это есть их индексы). Каждый раз домножаем результат на найденное значение.

Максимально исправить

Чтобы число было минимальным, необходимо идти с конца строки. Находим первый подходящий символ, который можно заменить на больший. Далее меняем все символы правее него, если это возможно на меньшие. Если не одной замены не произошло, выводим -1.

Пирамидки

Сортируем список. Пропускаем через множество. Таким образом, мы узнаем, сколько уникальных элементов исходного списка существует. Создаем словарь из такого количества элементов, они будут ключами. Затем сосчитаем количество встречаемости каждого элемента в исходном списке, заносим в словарь.

Проходим по словарю. Заносим количество самых младших элементов в результат. Проверяем, если предыдущий элемент был на единицу меньше текущего, то набор элементов продолжается, и нужно к результату прибавить значение только тогда, когда количество элементов нового соседнего типа больше, чем предыдущего. Если меньше, то ничего делать не нужно, эти наборы элементов просто заканчиваются.

А если предыдущий элемент отличается от последующего на значение большее единицы, то открываются новые наборы. Которые нужно добавить к результату и далее смотреть очередные элементы.