

IX открытый командный online турнир по
программированию
Юго-Восточного образовательного округа
Центр цифрового образования «ИТ-куб» на базе КОГОАУ
«Вятский многопрофильный лицей» г. Вятские Поляны
19 декабря 2025 года

Разбор задач

Кирпичный завод

Если внимательно изучить и поработать со входными данными и приводимыми примерами, можно увидеть, что для приготовления каждого облицовочного кирпича, необходимо использовать по 2 кг ЦПС. Следовательно, при заданном n ответом будет $2 * n$.

Нанобитва

В задаче достаточно подсчитать количество элементов, равных минимальному (только они не могут быть победителями, остальные могут) и вычесть это найденное значение из общего количества элементов.

Проверка Flash

Можно подсчитать количество единиц в исходной строке. Умножить это значение на количество символов в строке. Затем к этому значению добавить разницу между количеством нулей (именно каждый из них превратится в единицу) и количеством единиц (они превратятся в нули).

.

Контрольные произведения

Заведем два динамических массива, назовем их dp и do . В одном будем подсчитывать количество произведений, дающих положительный результат, в другом – отрицательный.

Первый элемент одного из массивов будет равен 1, другого - 0 в зависимости от знака исходного массива. Каждый элемент этого массива, начиная со следующего, будет вычисляться в зависимости от знака элемента исходного массива:

```
if a[i] > 0:
    dp[i] = dp[i-1] + 1
    do[i] = do[i-1]
else:
    dp[i] = do[i-1]
    do[i] = dp[i-1] + 1
```

Из формул видно, что на текущий элемент будет влиять отрицательный знак исходного элемента.

Результатом задачи будут суммы элементов этих массивов.

Две каретки

Проанализировав все возможные исходы развития ситуации, можно сделать следующие выводы.

Если первая каретка не может сделать ход, например, она находится в крайней позиции, а на соседней позиции штанги находится вторая каретка, то она проигрывает. Первая каретка проигрывает и в случае, когда количество кувшинок между ее начальной позицией и позицией второй каретки - четно. Она двигается первой, а вторая каретка дублирует ее ход, двигаясь навстречу. Первой каретке не останется позиций, чтобы выиграть.

Первая каретка выиграет только тогда, когда количество позиций между каретками нечетно, так как она всегда начинает двигаться первой, она займет последнюю свободную позицию, двигаясь навстречу второй каретке.

Чартер

В задаче гарантированно сказано, что каждое $l_i > r_{i-1}$. Это значит, что сортировку промежутков проводить не нужно.

Для решения достаточно проверить, чтобы именно эти промежутки $l_i - r_{i-1}$ были $\geq s$. Еще нужно проверить время до начала первого промежутка и время после окончания последнего промежутка.

Если хотя бы один из возможных промежутков подходит, то выводим положительный ответ, иначе отрицательный.

Стабильные астероиды

Числа необходимо отсортировать.

Находим количество получившихся групп астероидов и заносим в новый массив те разницы между соседними элементами, которые нас не устраивают, то есть, обрывают группу. Несложно догадаться, что элементов в нем будет на один меньше, чем количество групп.

Чтобы в итоге групп получилось, как можно меньше, необходимо оставшимися значениями заполнять сначала маленькие промежутки. Поэтому новый массив тоже сортируем.

Затем распределяем имеющиеся новые астероиды по промежуткам методом деления величины промежутка на допустимую разницу элементов в группе. При этом необходимо постоянно учитывать количество оставшихся нераспределенных новых астероидов.

Коллайдер

Считаем общее количество ж-фионов, которое нам потребуется. Их будет $y_e = 2 * x + y$, вычтем из этого количества число уже имеющихся, а это a . Если число будет отрицательным, то $y_e = 0$. Значит ж-фионы нам дополнительно синтезировать не нужно.

Аналогичные манипуляции проделываем и с подсчетом с-фионов. Их нужно $b_l = 3 * z + y - b$. Если число будет отрицательным, то $b_l = 0$.

Результат = $y_e + b_l$.

Помогите ЧПУ

Для начала нужно подсчитать количество команд каждого значения, поможет сортировка подсчетом. Находим сумму s всех значений команд.

Затем перебираем все возможные уникальные значения команд, назовем их x , которые могут остаться в коде. Для каждого значения x находим количество команд, которые могут быть удалены, то есть, находим сумму тех значений, которые $\geq x$.

Сумму удаленных команд можно подсчитать разницей между s и суммой выбранных значений. Среди этих значений находим минимальное, оно и будет ответом.

Три рулетки и треугольник

Несложно догадаться, что треугольник можно построить тогда, когда сумма двух меньших сторон хотя бы на единицу больше большей стороны. В противном случае, мы должны добиться истинности этого неравенства.

Ответом будет разница между большей стороной и суммой двух меньших сторон + 1.

Робоконтроль

Ограничения по данным позволяют на базе исходных массивов создать новые, выбрав туда все изделия, в которых хотя бы, одно не имеет отклонений.

Затем просматриваем все эти пары, на каждом шаге анализируя значения обоих массивов. Если оба изделия в паре не имеют отклонений, то учитываем его.

Если одно из изделий в паре имеет отклонение, то увеличиваем счетчик выбранных для упаковки изделий только, если не повторились значения предыдущей пары. Для подсчета можно создать список из такого же количества элементов, что и полученные при первой обработке массивы. И менять в этом списке значения, сохраняя информацию о том, какой элемент пары имеет отклонение по допуску.

Цепочки

Необходимо использовать дополнительную структуру, в которой будут храниться все просмотренные буквы. Пусть это будет список, назовем его `b`.

Проходим по исходной строке, если список `b` не пуст, сравниваем очередной символ строки с последним элементом, помещенным в список `b`.

Если они одинаковы, то удаляем из списка `b` последний элемент, иначе помещаем туда новый символ.

Конечное содержимое списка будет результатом. Необходимо вывести его в виде строки.

Досмотр интроскопом

Минимальное время досмотра считается просто: если высота i -го чемодана больше высоты интраскопа, то к счетчику добавляем 2, иначе добавляем 1.