

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Квазидвоичная сумма

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Число назовем *квазидвоичным*, если в его десятичной записи присутствуют только цифры 0 или 1. Например, числа 0, 1, 101, 110011 — квазидвоичные, а числа 2, 12, 900 — нет.

Вам дано положительное целое число n . Представьте его в виде суммы минимального количества квазидвоичных чисел.

Входные данные

В первой строке записано единственное целое число n ($1 \leq n \leq 10^6$).

Выходные данные

В первой строке выведите одно целое число k — минимальное количество чисел в представлении числа n в виде суммы квазидвоичных.

Во второй строке выведите k чисел — элементы суммы. Все эти числа должны быть квазидвоичными согласно определению выше, их сумма должна равняться n . Ведущие нули в числах выводить не нужно. Порядок чисел не имеет значения. Если существует несколько возможных представлений, разрешается вывести любое.

Примеры

входные данные	Скопировать
9	
выходные данные	Скопировать
9 1 1 1 1 1 1 1 1	
входные данные	Скопировать
32	
выходные данные	Скопировать
3 10 11 11	



[ЗАДАЧИ](#) [ОТΟΣЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Увеличения в группах

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дан массив a длины n . Чтобы вычислить штраф, нужно выполнить следующие действия:

1. Разбить массив a на две (возможно, пустых) подпоследовательности $^{\dagger} s$ и t такие, что каждый элемент a находится либо в s , либо в t .
2. Для массива b длины m определим штраф $p(b)$ массива b как количество индексов i между 1 и $m - 1$ таких, что $b_i < b_{i+1}$.
3. Общий штраф, который вы получите, равен $p(s) + p(t)$.

Найдите минимально возможный штраф, который вы получите, если выполните вышеописанные действия оптимально.

† Последовательность x является подпоследовательностью y , если x может быть получена из y удалением нескольких (возможно, ни одного или всех) элементов.

‡ Некоторыми допустимыми способами разбиения массива $a = [3, 1, 4, 1, 5]$ на (s, t) являются $([3, 4, 1, 5], [1])$, $([1, 1], [3, 4, 5])$ и $([], [3, 1, 4, 1, 5])$, в то время как некоторыми недопустимыми способами разбиения a являются $([3, 4, 5], [1])$, $([3, 1, 4, 1], [1, 5])$ и $([1, 3, 4], [5, 1])$.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длину массива a .

Вторая строка каждого набора содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — элементы массива a .

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число, представляющее собой минимальный возможный штраф, который вы можете получить.

Пример

входные данные	Скопировать
<pre> 5 5 1 2 3 4 5 8 8 2 3 1 1 7 4 3 5 3 3 3 3 3 1 1 2 2 1 </pre>	
выходные данные	Скопировать
<pre> 3 1 0 0 0 </pre>	

Примечание

В первом наборе входных данных возможный способ разбиения a — $s = [2, 4, 5]$ и $t = [1, 3]$. Штраф равняется $p(s) + p(t) = 2 + 1 = 3$.

Во втором наборе входных данных возможный способ разбиения a — $s = [8, 3, 1]$ и $t = [2, 1, 7, 4, 3]$. Штраф равняется $p(s) + p(t) = 0 + 1 = 1$.

В третьем наборе входных данных возможным способом разбиения a является $s = []$ и $t = [3, 3, 3, 3, 3]$. Штраф равняется $p(s) + p(t) = 0 + 0 = 0$.

ЗАДАЧИ

ОТОСЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

С. Собери команды

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

У вас есть n программистов, которых вы хотите распределить по командам. Навык i -го программиста равен a_i . Вы хотите собрать из них максимальное количество команд. Для команд есть одно ограничение: количество программистов в команде, умноженное на минимальный навык среди всех программистов этой команды, должно быть как минимум x .

Каждый программист может находиться максимум в одной команде. Некоторые программисты могут остаться без команды.

Посчитайте максимальное количество команд, которое вы можете собрать.

Входные данные

Первая строка содержит одно число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит два числа n и x ($1 \leq n \leq 10^5; 1 \leq x \leq 10^9$) — количество программистов и ограничение на навык команды соответственно.

Вторая строка каждого набора входных данных содержит n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), где a_i равно навыку i -го программиста.

Сумма n по всем наборам не превосходит 10^5 .

Выходные данные

На каждый набор входных данных выведите одно число — максимальное количество команд, которое вы можете собрать.

Пример

входные данные	Скопировать
3 5 10 7 11 2 9 5 4 8 2 4 2 3 4 11 1 3 3 7	
выходные данные	Скопировать
2 1 0	



ЗАДАЧИ

ОТОСЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

В. Орак и модели

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

В магазине есть n моделей, пронумерованных от 1 до n , размеры которых равны s_1, s_2, \dots, s_n .

Орак купит некоторые из этих моделей и упорядочит их по возрастанию номеров (индексов, а не размеров).

Орак считает, что полученная расстановка **красивая**, если для любых двух соседних моделей с номерами i_j и i_{j+1} (обратите внимание, что $i_j < i_{j+1}$, так как Орак упорядочил их правильно), i_{j+1} делится на i_j и $s_{i_j} < s_{i_{j+1}}$.

Например, для 6 моделей с размерами $\{3, 6, 7, 7, 7, 7\}$, он может купить модели с индексами 1, 2, и 6, и полученная расстановка будет красивой. Обратите внимание, что расстановка из одной модели также считается красивой.

Орак хочет знать, какое наибольшее число моделей он может купить, и он может задавать вам эти вопросы по несколько раз.

Входные данные

В первой строке записано одно целое число t ($1 \leq t \leq 100$): количество запросов.

Каждый запрос состоит из двух строк, в первой из которых записано одно целое число n ($1 \leq n \leq 100\,000$): количество моделей в магазине, а во второй записаны n целых чисел s_1, \dots, s_n ($1 \leq s_i \leq 10^9$): размеры моделей.

Гарантируется, что сумма величин n не превосходит 100 000.

Выходные данные

Выведите t строк, в i -й из которых должно быть записано максимальное число моделей, которое Орак может купить для i -го запроса.

Пример

входные данные

Скопировать

4
4
5 3 4 6
7
1 4 2 3 6 4 9
5
5 4 3 2 1
1
9

выходные данные

Скопировать

2
3
1
1

Примечание

Для первого запроса, например, Орак может купить модели с индексами 2 и 4, расстановка которых будет красивой так как 4 делится на 2 и 6 больше, чем 3. Рассмотрев остальные варианты, можно легко убедиться, что нет красивой расстановки с более, чем тремя моделями.

Во втором запросе Орак может купить модели с индексами 1, 3, и 6. Рассмотрев остальные варианты, можно легко убедиться, что нет красивой расстановки с более, чем тремя моделями.

В третьем примере не существует красивой расстановки с более, чем одной моделью.

Codeforces

(с) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 04.11.2024 20:30:51^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке

<https://codeforces.com/problemset/problem/1350/B>

1/2

В. Машмох и ACM

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Бимоху, начальнику Машмоха, Машмох не нравился. Вот он его и уволил. Решил тогда Машмох новую работу не искать, а поступить в университет и поучаствовать в ACM. Машмох хочет попасть в команду Бамоха. Для этого ему дали (в качестве испытания) несколько задач по программированию и неделю на их решение. Машмох не шибко умудренный программист. В общем-то, он и не программист вовсе. Так что ничего он не решил, а попросил вас помочь ему с этими заданиями. Одно из них такое:

Последовательность из l целых чисел b_1, b_2, \dots, b_l ($1 \leq b_1 \leq b_2 \leq \dots \leq b_l \leq n$) называется *хорошей*, если каждое число делит без остатка следующее число в последовательности. Более формально, $b_i \mid b_{i+1}$ для всех i ($1 \leq i \leq l - 1$).

Вам даны n и k , найдите количество хороших последовательностей длины k . Так как ответ может быть достаточно большим, выведите его по модулю 1000000007 ($10^9 + 7$).

Входные данные

В первой строке записано два целых числа через пробел n, k ($1 \leq n, k \leq 2000$).

Выходные данные

Выведите единственное целое число — количество хороших последовательностей длины k по модулю 1000000007 ($10^9 + 7$).

Примеры

входные данные	Скопировать
3 2	
выходные данные	Скопировать
5	

входные данные	Скопировать
6 4	
выходные данные	Скопировать
39	

входные данные	Скопировать
2 1	
выходные данные	Скопировать
2	

Примечание

В первом примере хорошие последовательности такие: [1, 1], [2, 2], [3, 3], [1, 2], [1, 3].

Codeforces

(с) Copyright 2010-2024 Михаил Мирзаянов


Соревнования по программированию 2.0


Время на сервере: 04.11.2024 20:30:49^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке





https://codeforces.com/problemset/problem/414/B

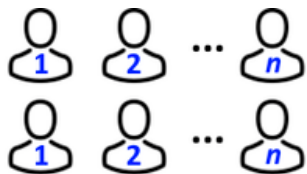
1/2

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Баскетбольная зарядка

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

В ЛКШ наконец-то появилось поле для игры в баскетбол, и на радостях Демид решил провести баскетбольную зарядку. На зарядку к Демиду пришло $2 \cdot n$ человек, и он построил их в два ряда по n человек в каждом. В каждом из рядов он пронумеровал игроков от 1 до n слева направо.



Теперь Демид хочет выбрать команду для игры в баскетбол. Он будет выбирать игроков слева направо, и номер каждого следующего **взятого** игрока будет строго больше, чем предыдущего взятого. А для того, чтобы не отдавать предпочтения одному из рядов, каждый следующий выбранный школьник должен стоять не в том же ряду, что предыдущий. Первый выбранный школьник может быть любым из всех $2n$, а количество игроков в команде не ограничено.

Демид считает, что команда тем лучше, чем больше суммарный рост ее игроков. Помогите Демиду определить максимальный суммарный рост игроков команды, которую он может выбрать.

Входные данные

В первой строке дано число n ($1 \leq n \leq 10^5$) — количество школьников в каждом из рядов.

Вторая строка входных данных содержит n целых чисел $h_{1,1}, h_{1,2}, \dots, h_{1,n}$, разделенных пробелами ($1 \leq h_{1,i} \leq 10^9$), где $h_{1,i}$ равно росту i -го человека в первом ряду.

Третья строка входных данных содержит n целых чисел $h_{2,1}, h_{2,2}, \dots, h_{2,n}$, разделенных пробелами ($1 \leq h_{2,i} \leq 10^9$), где $h_{2,i}$ равно росту i -го человека во втором ряду.

Выходные данные

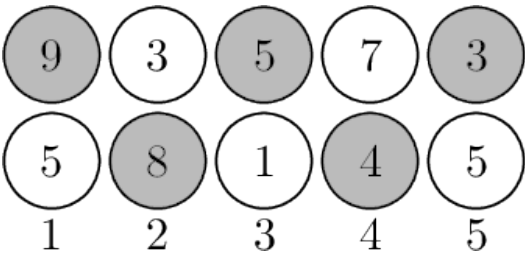
Выведите одно число — максимальное суммарный рост игроков в команде, которую может выбрать Демид.

Примеры

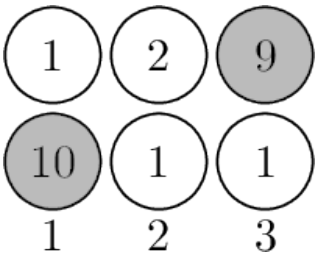
входные данные	Скопировать
5 9 3 5 7 3 5 8 1 4 5	
выходные данные	Скопировать
29	
входные данные	Скопировать
3 1 2 9 10 1 1	
выходные данные	Скопировать
19	
входные данные	Скопировать
1 7 4	
выходные данные	Скопировать
7	

Примечание

В первом примере Демид может выбрать такую команду:



Во втором примере Демид может выбрать такую команду:



[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 04.11.2024 20:30:46^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



CODEFORCES

Sponsored by TON

UK

RU

Кар6502 | Выйти

ЗАДАЧИОТОСЛАТЬСТАТУСПОЛОЖЕНИЕЗАПУСК

А. Каникулы

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Каникулы Васи будут длиться n дней! И Вася решил за это время улучшить свои навыки программирования, а также позаниматься спортом. Про каждый из n дней каникул Вася знает, будет ли открыт в этот день спортзал, а также будет ли в интернете в этот день проводиться конкурс. Для i -го дня возможны четыре варианта:

- 1. в этот день закрыт спортзал и не проводится конкурс;
- 2. в этот день закрыт спортзал и проводится конкурс;
- 3. в этот день открыт спортзал и не проводится конкурс;
- 4. в этот день открыт спортзал и проводится конкурс.

В каждый из дней Вася может либо отдыхать, либо писать конкурс (если он проводится в этот день), либо заниматься спортом (если открыт спортзал).

Перед вами стоит задача найти минимальное количество дней, в которые Вася будет отдыхать (то есть не будет заниматься спортом и не будет писать конкурс одновременно). Единственное ограничение от Васи — *он не хочет два дня подряд заниматься одним и тем же видом активности, то есть он не будет заниматься спортом два дня подряд и он не будет писать конкурсы два дня подряд*.

Входные данные

В первой строке следует целое положительное число n ($1 \leq n \leq 100$) — количество дней в каникулах Васи.

Во второй строке следует через пробел последовательность целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 3$), где:

- a_i равно 0, если в i -й день каникул не работает спортзал и не проводится конкурс;
- a_i равно 1, если в i -й день каникул не работает спортзал, но проводится конкурс;
- a_i равно 2, если в i -й день каникул работает спортзал и не проводится конкурс;
- a_i равно 3, если в i -й день каникул работает спортзал и проводится конкурс.

Выходные данные

Выведите минимально возможное количество дней, в которые Вася будет отдыхать. Помните, что Вася отказывается:

- заниматься спортом в какие-либо два подряд идущих дня,
- писать конкурс в какие-либо два подряд идущих дня.

Примеры

входные данные	Скопировать
4 1 3 2 0	
выходные данные	Скопировать
2	

входные данные	Скопировать
7 1 3 3 2 1 2 3	
выходные данные	Скопировать
0	

входные данные	Скопировать
2 2 2	
выходные данные	Скопировать
1	

Примечание

В первом тестовом примере Вася может написать конкурс в день номер 1 и позаниматься спортом в день номер 3. Таким образом, он будет отдыхать всего два дня.

В. Я ненавижу 1111

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Вам дано целое число x . Можете ли вы получить x , просуммировав некоторое количество 11, 111, 1111, 11111, ...? (Вы можете использовать любое число среди них любое количество раз).

Например,

- $33 = 11 + 11 + 11$
- $144 = 111 + 11 + 11 + 11$

Входные данные

Первая строка ввода содержит одно целое число t ($1 \leq t \leq 10000$) — количество наборов входных данных.

Первая и единственная строка каждого набора входных данных содержит одно целое число x ($1 \leq x \leq 10^9$) — число, которое вы должны получить.

Выходные данные

Для каждого набора входных данных вы должны вывести одну строку. Если вы можете получить x , выведите «YES» (без кавычек). В противном случае выведите «NO».

Вы можете вывести каждую букву из «YES» и «NO» в любом регистре (верхнем или нижнем).

Пример

входные данные	Скопировать
3 33 144 69	
выходные данные	Скопировать
YES YES NO	

Примечание

Способы получения 33 и 144 были представлены в условии. Можно показать, что мы не можем представить 69 таким образом.

При поддержке



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Даны длина и сумма цифр...

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Вам задано положительное целое число m и неотрицательное целое число s . Ваша задача найти наименьшее и наибольшее из чисел, которые имеют длину m и сумму цифр s . Искомые числа должны быть неотрицательными целыми, записанными в десятичной системе счисления без ведущих нулей.

Входные данные

В единственной строке входных данных записана пара целых чисел m, s ($1 \leq m \leq 100, 0 \leq s \leq 900$) — длина и сумма цифр искомых чисел.

Выходные данные

В выходные данные выведите пару искомых неотрицательных целых чисел — сначала минимальное из возможных, потом — максимальное. Если ни одного числа, удовлетворяющего условию, не существует, то выведите пару чисел «-1 -1» (без кавычек).

Примеры

входные данные	Скопировать
2 15	
выходные данные	Скопировать
69 96	
входные данные	Скопировать
3 0	
выходные данные	Скопировать
-1 -1	



F. Очередная задача про пары, удовлетворяющие неравенству

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам дан массив целых чисел a_1, a_2, \dots, a_n . Найдите число пар индексов $1 \leq i, j \leq n$ таких, что $a_i < i < a_j < j$.

Входные данные
Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных. Далее следует их описание.

Первая строка каждого набора содержит число n ($2 \leq n \leq 2 \cdot 10^5$) — длину массива.

Вторая строка каждого набора содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — элементы массива.

Гарантируется, что сумма n по всем наборам не превосходит $2 \cdot 10^5$.

Выходные данные
Для каждого набора входных данных выведите одно число — количество пар индексов, удовлетворяющих неравенству из условия.

Пожалуйста, обратите внимание, что ответ для некоторых тестовых примеров может не поместиться в 32-разрядный целочисленный тип, поэтому вы должны использовать по крайней мере 64-разрядный целочисленный тип в вашем языке программирования (например, `long long` для C++).

Пример

входные данные	Скопировать
5 8 1 1 2 3 8 2 1 4 2 1 2 10 0 2 1 6 3 4 1 2 8 3 2 1 1000000000 3 0 1000000000 2	
выходные данные	Скопировать
3 0 10 0 1	

Примечание
В первом наборе входных данных пары $(i, j) = \{(2, 4), (2, 8), (3, 8)\}$.

- Пара $(2, 4)$ подходит, потому что $a_2 = 1, a_4 = 3$ и $1 < 2 < 3 < 4$.
- Пара $(2, 8)$ подходит, потому что $a_2 = 1, a_8 = 4$ и $1 < 2 < 4 < 8$.
- Пара $(3, 8)$ подходит, потому что $a_3 = 2, a_8 = 4$ и $2 < 3 < 4 < 8$.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Найди различные!

ограничение по времени на тест: 5 секунд
 ограничение по памяти на тест: 256 мегабайт

Вам дан массив a из n целых чисел, а также q запросов.

Каждый запрос представлен двумя числами l и r ($1 \leq l \leq r \leq n$). Ваша задача для каждого запроса найти такие два индекса i и j (или сообщить, что их не существует), что:

- $l \leq i \leq r$;
- $l \leq j \leq r$;
- $a_i \neq a_j$.

Иными словами, вам нужно для каждого запроса найти пару различных элементов среди a_l, a_{l+1}, \dots, a_r или сообщить, что такой пары не существует.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют описания наборов.

Первая строка каждого набора содержит одно целое число n ($2 \leq n \leq 2 \cdot 10^5$) — длину массива a .

Вторая строка каждого набора содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$) — элементы массива a .

Третья строка каждого набора содержит одно целое число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов.

Следующие q строк содержат по два целых числа l и r ($1 \leq l < r \leq n$) — границы запроса.

Гарантируется, что сумма значений n по всем наборам не превышает $2 \cdot 10^5$. Аналогично, гарантируется, что сумма значений q по всем наборам не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого запроса выведите два числа: искомые i и j ($l \leq i, j \leq r$), для которых $a_i \neq a_j$. Если такой пары не существует, выведите $i = -1$ и $j = -1$.

Вы можете разделять выводы для наборов входных данных пустыми строками. Это не является обязательным требованием.

Пример

входные данные	Скопировать
<pre> 5 5 1 1 2 1 1 3 1 5 1 2 1 3 6 30 20 20 10 10 20 5 1 2 2 3 2 4 2 6 3 5 4 5 2 3 4 4 1 2 1 4 2 3 2 4 5 1 4 3 2 4 5 1 5 2 4 3 4 3 5 </pre>	

4 5
5
2 3 1 4 2
7
1 2
1 4
1 5
2 4
2 5
3 5
4 5

выходные данные

Скопировать

2 3
-1 -1
1 3

2 1
-1 -1
4 2
4 6
5 3

1 2
1 2
2 3
3 2

1 3
2 4
3 4
5 3
5 4

1 2
4 2
1 3
2 3
3 2
5 4
5 4

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 04.11.2024 20:30:21^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Ряды монет

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Алиса и Боб играют в игру на матрице, состоящей из 2 строк и m столбцов. В ячейке в i -й строке в j -м столбце лежит $a_{i,j}$ монет.

Изначально Алиса и Боб стоят в ячейке $(1, 1)$. Они собираются пройти какой-то последовательностью ходов в ячейку $(2, m)$.

Возможные ходы следующие:

- Пойти направо — из некоторой ячейки (x, y) в $(x, y + 1)$;
- Пойти вниз — из некоторой ячейки (x, y) в $(x + 1, y)$.

Сначала Алиса делает **все свои ходы**, пока не достигнет $(2, m)$. Она собирает монеты во всех клетках, которые они посещает (включая стартовую).

Когда Алиса заканчивает, Боб начинает свой путь. Он также делает ходы, чтобы достичь $(2, m)$, и собирает монеты во всех клетках, которые он посетил, а **Алиса нет**.

Счет в игре равен суммарному количеству монет, которые собрал Боб.

Алиса хочет минимизировать счет. Боб хочет максимизировать счет. Какой будет счет в игре, если оба игрока играют оптимально?

Входные данные

В первой строке записано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Затем следуют описания t наборов входных данных.

В первой строке каждого набора записано одно целое число m ($1 \leq m \leq 10^5$) — количество столбцов в матрице.

В i -й из следующих 2 строк записаны m целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($1 \leq a_{i,j} \leq 10^4$) — количество монет в ячейке в i -м ряду в j -м столбце матрицы.

Сумма m по всем наборам входных данных не превосходит 10^5 .

Выходные данные

На каждый набор входных данных выведите одно целое число — счет игры, если оба игрока играют оптимально.

Пример

входные данные	Скопировать
<pre>3 3 1 3 7 3 5 1 3 1 3 9 3 5 1 1 4 7</pre>	
выходные данные	Скопировать
<pre>7 8 0</pre>	

Примечание

Пути для наборов входных данных из примеров изображены ниже. Путь Алисы раскрашен красным, а путь Боба — синим.

CODEFORCES

Sponsored by TON

|





[Кар6502](#) | [Выйти](#)

ЗАДАЧИ

ОТОСЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

А. Странный день рождения

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

На свой день рождения Петя пригласил n друзей, у i -го друга есть параметр k_i . Каждому из приглашенных друзей Петя хочет подарить ровно один подарок. В магазине есть m различных подарков, которые Петя может купить, каждый подарок можно купить **не** более одного раза. Подарок номер j стоит c_j рублей ($1 \leq c_1 \leq c_2 \leq \dots \leq c_m$).

Для i -го друга Петя может либо купить ему подарок $j \leq k_i$, что будет стоить Пете c_j рублей, либо заплатить другу c_{k_i} рублей вместо подарка. Помогите Пете определить наименьшую возможную стоимость проведения праздника.

Входные данные

В первой строке входных данных дано целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных.

Каждый набор входных данных начинается с целых чисел n и m ($1 \leq n, m \leq 3 \cdot 10^5$) — количество друзей у Пети и количество подарков.

Во второй строке набора даны n целых чисел k_1, k_2, \dots, k_n ($1 \leq k_i \leq m$) — параметры друзей Пети.

В третьей строке набора входных данных даны m целых чисел c_1, c_2, \dots, c_m ($1 \leq c_1 \leq c_2 \leq \dots \leq c_m \leq 10^9$) — стоимости подарков.

Гарантируется, что сумма n по всем тестам не превосходит $3 \cdot 10^5$, и что сумма m по всем тестам не превосходит $3 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите минимальное число рублей, которое понадобится Пете для приобретения всех подарков.

Примеры

входные данные	Скопировать
2 5 4 2 3 4 3 2 3 5 12 20 5 5 5 4 3 2 1 10 40 90 160 250	
выходные данные	Скопировать
30 190	

входные данные	Скопировать
1 1 1 1 1	
выходные данные	Скопировать
1	

Примечание

В примере есть два набора входных данных. В первом у Пети есть 5 друзей и 4 возможных подарка. Ответ 30 достигается, если подарить:

- 5 рублей первому другу.
- подарок стоимости 12 рублей второму другу.
- подарок стоимости 5 рублей третьему другу.
- подарок стоимости 3 рубля четвертому другу.
- 5 рублей пятому другу.

Во втором у Пети есть 5 друзей и 5 возможных подарков. Ответ 190 достигается, если подарить:

- подарок стоимости 10 рублей первому другу.

В. Вкусные покупки

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Сегодня Яссер и Адель пошли в магазин за пирожными. На полке в магазине стоят n типов пирожных, пронумерованные от 1 до n . Количество пирожных каждого типа бесконечно. Определим вкус i -го пирожного некоторой величиной a_i , которая является целым числом. В магазине есть как вкусные пирожные, так и неприятные, поэтому данная величина может быть положительной, нулем или отрицательной.

Яссер, разумеется, хочет попробовать их все, поэтому он купит ровно по одному пирожному каждого типа.

Адель, с другой стороны, выберет некоторый отрезок $[l, r]$ ($1 \leq l \leq r \leq n$), который содержит не все пирожные (он не может выбрать $[l, r] = [1, n]$) и купит ровно по одному пирожному каждого из типов $l, l + 1, \dots, r$.

После этого они сравнят суммарный вкус купленных ими пирожных. Яссер будет счастлив, если суммарный вкус пирожных, которые он купил, **строго** больше, чем суммарный вкус пирожных, которые купил Адель **вне зависимости от выбора Адель**.

Например, пусть вкус пирожных будет $[7, 4, -1]$. Яссер купит все, суммарный вкус будет равен $7 + 4 - 1 = 10$. Адель может выбрать отрезки $[7]$, $[4]$, $[-1]$, $[7, 4]$ или $[4, -1]$, суммарный вкус будет равен $7, 4, -1, 11$ и 3 , соответственно. Адель может выбрать отрезок с суммарным вкусом 11 , и раз 10 не строго больше 11 , то Яссер не будет счастлив :(

Узнайте, будет ли Яссер счастлив после похода в магазин.

Входные данные

В каждом тесте содержится несколько наборов входных данных. В первой строке записано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Затем следует описание наборов входных данных.

В первой строке каждого набора входных данных записано одно целое число n ($2 \leq n \leq 10^5$).

Во второй строке каждого набора входных данных записаны n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$), где a_i является вкусом i -го типа пирожного.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 .

Выходные данные

Для каждого набора входных данных выведите «YES», если суммарный вкус пирожных, которые купил Яссер, **строго** больше, чем суммарный вкус пирожных, которые купил Адель **вне зависимости от выбора Адель**. В противном случае выведите «NO».

Пример

входные данные	Скопировать
<pre>3 4 1 2 3 4 3 7 4 -1 3 5 -5 5</pre>	
выходные данные	Скопировать
<pre>YES NO NO</pre>	

Примечание

В первом примере суммарный вкус любого отрезка, который может выбрать Адель, меньше, чем суммарный вкус всех пирожных.

Во втором примере Адель выберет отрезок $[1, 2]$ с суммарным вкусом 11 , что не меньше, чем суммарный вкус всех пирожных, который равен 10 .

В третьем примере, Адель может выбрать отрезок $[3, 3]$ с суммарным вкусом 5 . Обратите внимание, что вкус пирожных у Яссера тоже равен 5 , поэтому в данном случае суммарный вкус пирожных, которые купил Яссер, не строго больше, чем суммарный вкус пирожных, которые купил Адель.

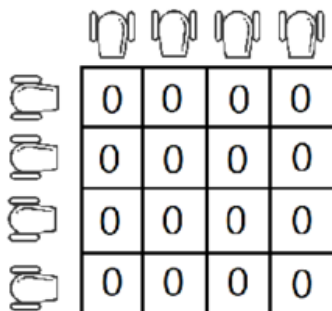
ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

Е. Полигон

ограничение по времени на тест: 2 секунды
 ограничение по памяти на тест: 256 мегабайт

Полигон — это не только лучшая платформа для разработки задач, но и квадратная матрица со стороной n , изначально состоящая из нулей.

На полигоне проводятся боевые учения. Поэтому над каждой клеткой в первой строке и слева от каждой клетки первого столбца находится пушка. Таким образом, всего есть $2n$ пушек.



Изначальный полигон для $n = 4$.

Пушки стреляют единицами. В один момент времени стреляет не больше одной пушки. Когда единица вылетает из пушки, то она летит вперед, по направлению выстрела, до тех пор, пока не столкнется с границей полигона или другой единицей. После этого она занимает клетку, в которой находилась перед столкновением, и остается там. Изучите примеры для лучшего понимания.

Более формально:

- если пушка, стоящая в строке i перед первым столбцом, стреляет единицей, то единица начинает свой полет из клетки $(i, 1)$ и заканчивает в какой-то клетке (i, j) ;
- если пушка, стоящая в столбце j над первой строкой, стреляет единицей, то единица начинает свой полет из клетки $(1, j)$ и заканчивает в какой-то клетке (i, j) .

Например, рассмотрим следующую последовательность выстрелов:



У вас на столе лежит отчет с проведенных учений. Этот отчет является квадратной матрицей с длиной стороны n , состоящей из нулей и единиц. Вам интересно, действительно ли произошли учения. Другими словами, существует ли такая последовательность выстрелов, что в конце получится заданная матрица?

Каждая пушка может сделать произвольное количество выстрелов. Перед началом учений полигон состоит из нулей.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов тестовых данных в тесте. Далее следуют t наборов тестовых данных.

Каждый набор начинается со строки, в которой записано целое число n ($1 \leq n \leq 50$) — размер полигона.

Далее следуют n строк длины n , состоящих из нулей и единиц — матрица полигона после проведения учений.

Суммарная площадь матриц во всех наборах тестовых данных в одном тесте не превосходит 10^5 .

Выходные данные

Для каждого набора тестовых данных выведите:

- YES, если существует последовательность выстрелов, приводящая к заданной матрице;
- NO, если такой последовательности не существует.

Буквы в словах YES и NO можно выводить в любом регистре.

Пример

входные данные	Скопировать
5 4 0010 0011 0000 0000 2 10 01 2 00 00 4 0101 1111 0101 0111 4 0100 1110 0101 0111	
выходные данные	Скопировать
YES NO YES YES NO	

Примечание

Первый набор тестовых данных примера разобран в условии.

Ответ на второй набор NO, так как, вылетев из любой пушки, единица в клетке (1, 1) продолжила бы свой полет дальше.

