

## [C] 1. Сложный контекст

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В программе параллели F есть  $M$  различных тем. Преподаватели составляют контекст из  $N$  задач. У каждой задачи есть ровно одна тема, при этом она может не входить в программу. В зависимости от темы задаче присваивается сложность. Суммарная сложность контекста – это сумма сложностей всех задач. В контексте может быть как несколько задач на одну тему, так и не быть задач на какую-то тему вовсе.

У темы  $j$  есть название – строка  $Q_j$ , и сложность соответствующих задач –  $D_j$ . Напротив каждой из  $N$  задач в контексте написана ее тема, для  $i$ -й задачи это строка  $T_i$ . Если  $T_i$  входит в программу под номером  $j$  (то есть  $T_i = Q_j$ ), то сложность задачи будет  $D_j$ . Если же  $T_i$  отсутствует в программе, то сложность задачи равна константе  $D_0$ .

По заданной программе, сложностям тем и списку задач в контексте определите суммарную сложность всех задач в контексте.

### Формат входных данных

Первая строка содержит два целых числа  $N$  ( $1 \leq N \leq 100$ ) и  $M$  ( $1 \leq M \leq 100$ ) – количество задач в контексте и количество тем в программе соответственно.

Вторая строка содержит  $N$  строк  $T_1, T_2, \dots, T_N$ , записанных через пробел. Строки состоят из маленьких латинских букв и не превышают в длину 20 символов.

Третья строка содержит  $M$  строк  $Q_1, Q_2, \dots, Q_M$ , записанных через пробел. Строки состоят из маленьких латинских букв и не превышают в длину 20 символов.

Четвертая строка содержит  $M + 1$  натуральное число  $D_0, D_1, D_2, \dots, D_M$  ( $1 \leq D_i \leq 1000$ ).

### Формат выходных данных

Выведите единственное натуральное число – суммарную сложность всех задач в контексте.

### Пример

стандартный ввод	стандартный вывод
4 2 sort realisation map sort map sort 13 1 5	24

### Замечание

В примере из условия в программе 2 темы, а в контексте 4 задачи, причем темы «realisation» нет в программе. Тогда мы получаем сложность 13 за «realisation», за две задачи по теме «sort» суммарно получаем тоже 10 (как  $5 \times 2$ ) и за задачу по теме «map» получаем 1. Суммарно 24.

## [C] 2. Очередная игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Тимофей и Сережа не стали слушать разбор задач, так как все решили сами, поэтому им пришлось придумать себе занятие. Они решили играть в следующую игру. Сперва каждый из двух игроков независимо придумывает по натуральному числу. Пусть Тимофей выбрал число  $A$ , а Сережа выбрал число  $B$ . Затем по этим двум числам итеративно считается величина  $X$ , изначально равная нулю: пока  $A \neq B$  мы либо делаем  $B = B - A$ , если  $B > A$ , и прибавляем к итоговой величине  $X$  единицу, либо делаем  $A = A - B$ , если  $A > B$ , и опять прибавляем к итоговой величине  $X$  единицу. В игре выигрывает Тимофей, если после этих вычислений получилось число более 20232024, иначе – Сережа.

Тимофей хочет подготовиться к любому исходу, поэтому просит вас написать программу, которая по заданным числам  $A$  и  $B$  найдет  $X$ .

Заметьте, что если при вычислении  $X$  достигает 20232024, то вычисление продолжается, а не прерывается.

### Формат входных данных

В единственной строке даны два натуральных числа, записанных через пробел –  $A$  и  $B$  ( $1 \leq A \leq 10^{18}, 1 \leq B \leq 10^{18}$ ).

### Формат выходных данных

Выведите единственное натуральное число – искомую величину  $X$ .

### Примеры

стандартный ввод	стандартный вывод
12 5	5
9 9	0
1000000000000 1	999999999999

### Замечание

Заметьте, что для хранения чисел  $A$  и  $B$  требуются 64-битные типы данных.

В первом примере  $A = 12, B = 5$ . Посмотрим, как будет формироваться величина  $X$  для этого примера.

- $A = 12, B = 5, A > B \rightarrow A = 7, B = 5, X = 1$
- $A = 7, B = 5, A > B \rightarrow A = 2, B = 5, X = 2$
- $A = 2, B = 5, B > A \rightarrow A = 2, B = 3, X = 3$
- $A = 2, B = 3, B > A \rightarrow A = 2, B = 1, X = 4$
- $A = 2, B = 1, A > B \rightarrow A = 1, B = 1, X = 5$

Получаем, что в конце  $X = 5$ .

Во втором примере мы сразу имеем  $A = B$ , поэтому  $X = 0$ .

## [C] 3. Последовательность

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дано целое нечетное число  $n$  и последовательность  $a_1, a_2, \dots, a_n$ .

Назовем последовательность красивой, если для каждого четного  $i$  верно, что  $a_{i-1} < a_i > a_{i+1}$ , например последовательность  $(1, 3, 1, 5, 2)$  является красивой, а последовательности  $(3, 1, 3)$ ,  $(1, 2, 2)$ ,  $(1, 3, 5)$  — нет. Вам необходимо понять, можно ли сделать из данной Вам последовательности красивую перестановкой элементов.

### Формат входных данных

В первой строке дано единственное число  $n$  ( $1 \leq n \leq 10^5$ ). Во второй строке дано  $n$  чисел —  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

В единственной строке выведите «Yes», если это возможно — «No» иначе.

### Примеры

стандартный ввод	стандартный вывод
5 1 3 5 7 9	Yes
5 2 3 3 3 2	No

### Замечание

В первом примере можно получить красивую последовательность следующего вида:  $(1, 7, 5, 9, 3)$ .

Во втором примере получить красивую последовательность не получится, так как на четных позициях должны стоять тройки, а их больше, чем четных позиций.

## [C] 4. Неоновые вывески

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Лёша любит гулять по городу. Во время прогулки он всегда обращает внимание на вывески всяких интересных заведений. Больше всего ему нравятся неоновые.

Сегодня Лёша в очередной раз проходил мимо одной из своих любимых вывесок. Время не щадит никого, поэтому некоторые буквы в вывеске погасли и не подсвечивались. Лёша заметил одну интересную особенность: если разбить вывеску на подотрезки подряд идущих горящих букв и подряд идущих негорящих букв (например, если в вывеске «Семёрочка» не горят буквы «м», «ё», «ч», «к», то она разбивается на 5 подотрезков: «Се», «мё», «ро», «чк», «а»), то каждый следующий подотрезок лексикографически больше предыдущего.

В этой задаче будем считать, что все строки состоят только из строчных латинских букв. Строка  $s$  длины  $m$  называется *лексикографически больше* строки  $t$  длины  $l$ , если выполнено одно из двух:

- 1) Существует  $1 \leq k \leq l$  такое, что  $s_1 = t_1, s_2 = t_2, \dots, s_{k-1} = t_{k-1}$ , а  $s_k$  идёт в алфавите позже  $t_k$ .
- 2)  $s_1 = t_1, s_2 = t_2, \dots, s_l = t_l$  и  $l < m$ .

Теперь проходя мимо каждой из  $n$  полностью подсвеченных неоновых вывесок на сегодняшнем маршруте, Лёша задавался вопросом: а можно ли погасить в вывеске несколько (**не меньше одной, но не все**) букв так, чтобы подстроки из подряд идущих горящих и негорящих букв шли в возрастающем лексикографическом порядке?

### Формат входных данных

В первой строке входных данных задано единственное целое число  $n$  ( $2 \leq n \leq 2000$ ) — количество вывесок.

В последующих  $2n$  строках описаны сами вывески. В первой строке для  $i$ -й вывески записано одно целое число  $k_i$  ( $2 \leq k_i \leq 2000$ ) — длина  $i$ -й вывески. В следующей строке задана сама вывеска  $s_i$  — строка, состоящая из строчных латинских букв. Суммарная длина всех вывесок не превосходит 2000.

### Формат выходных данных

Для каждой вывески выведите «Yes», если можно погасить несколько букв в ней так, чтобы удовлетворить условиям Лёши и «No» иначе.

### Пример

стандартный ввод	стандартный вывод
4	No
3	Yes
bab	No
4	Yes
abac	
5	
yriba	
12	
abababaaabbb	

### Замечание

В первом тестовом примере какие бы буквы мы не погасили, строка разобьётся либо на подотрезки «b» и «ab», либо на подотрезки «ba» и «b», либо на подотрезки «b», «a» и «b». В любом случае первый подотрезок всегда лексикографически больше второго, поэтому ответ — «No». Обратите внимание, что вы должны погасить **хотя бы одну** букву в вывеске, но **не все**.

Во втором тестовом примере можно погасить первую букву «a» и букву «c».

## [C] 5. Охрана территории

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Для охраны детского лагеря от различных угроз было решено выставить наблюдателей. Территория была поделена на квадраты и представляет из себя сетку  $a$  на  $b$ . Наблюдатель может стоять в любой точке данной сетки (хоть внутри квадратов, хоть на границе) и наблюдать за всеми квадратами, внутри которых находится данная точка или которых касается (например, для сетки  $2 \times 2$  наблюдатель стоящий ровно посередине просматривает все 4 квадрата).

Помогите подсчитать минимальное число наблюдателей, чтобы просматривать всю территорию лагеря.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq 1000$ ).

### Формат выходных данных

Выведите единственное число — ответ на задачу

### Примеры

стандартный ввод	стандартный вывод
2 2	1
3 3	4

### Замечание

В первом примере можно разместить одного наблюдателя посередине сетки

Во втором примере необходимо разместить 4 наблюдателей в узлах центрального квадрата. Меньше разместить не получится, так как на границе каждого из 4 угловых квадратов должен быть хотя бы один наблюдатель, при этом данные границы не пересекаются

## [C] 6. Базовый завтрак

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Миша делает базовый бутерброд себе на завтрак. Из чего состоит базовый бутерброд? Всё верно: из кусочка хлеба, ломтиков ветчины и сыра, а также кусочка помидора. Все 4 ингредиента Миша нарезал прямоугольниками.

К сожалению, Миша сегодня встал не с той ноги, поэтому он случайно уронил все ингредиенты на пол. Но он не расстроился, а представил пол как Декартову систему координат, обозначил координаты каждого прямоугольника-ингредиента и задался следующим вопросом: какую суммарную площадь заняли ингредиенты на полу?

### Формат входных данных

Входные данные состоят из четырёх строк, в каждой строке записано 4 целых числа, разделённых пробелом — координаты левого нижнего и правого верхнего угла соответствующего ингредиента. Все координаты по модулю не превосходят 1000.

### Формат выходных данных

Выведите единственное целое число — суммарную площадь пола, покрытую ингредиентами.

### Примеры

стандартный ввод	стандартный вывод
3 1 6 3 1 2 4 7 2 4 6 8 5 2 7 6	35
-1 -1 1 0 -1 -1 0 1 0 -1 1 1 -1 0 1 1	4

### Замечание

Обратите внимание, что если два или более ингредиентов покрывают один и тот же кусочек пола, то соответствующая площадь учитывается в сумме **только один раз**.

## [C-B'] 1. Заряд телефона

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Миша купил себе новый YouPhone. После очередного обновления в системе пропал счетчик процентов зарядов и пользоваться им стало очень сложно. Миша подробно изучал характеристики перед покупкой, поэтому он знает, что:

- с 100% до 0% телефон разряжается ровно за  $a$  минут, при этом он теряет заряд равномерно
- с 0% до 100% телефон заряжается ровно за  $b$  минут, при этом зарядка так же идет равномерно
- зарядка не может опускаться ниже 0% или становиться выше 100%

В течение дня Миша постоянно перемещался и его телефон менял источник энергии  $n$  раз (с зарядки от сети на аккумулятор и наоборот). Известно, что в начале дня (в 00:00) телефон стоял на зарядке и был полностью заряжен. Помогите Мише найти оставшийся в конце дня процент заряда на его телефоне

### Формат входных данных

Первая строка входного файла содержит числа  $n, a, b$  ( $0 \leq n \leq 20, 1 \leq a, b \leq 1500$ ) — количество смен источника энергии телефона, время разряда и заряда аккумулятора соответственно.

Каждая из следующих  $n$  строк содержит время суток в формате «hh:mm», в которое была произведена смена источника питания. Данные уникальны и перечислены в хронологическом порядке

### Формат выходных данных

В выходной файл выведите процент заряда батареи в 23:59 с точностью не менее четырех знаков после запятой.

### Примеры

стандартный ввод	стандартный вывод
4 337 326 05:54 08:34 15:30 20:57	58.79558
5 343 351 08:58 09:00 09:57 19:00 19:24	0.00000

## [C-B'] 2. Хулиганство

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Во время лекции кто-то из учеников перехватил управление доской и начал писать какие-то непонятные числа. Оказалось, что числа были далеко не случайные. Алиса заметила, что написанные на доске  $2N$  чисел разбивались на пары так, что для каждого числа  $X$  находилось число  $\alpha(X)$  такое, что  $\alpha(X)$  равно наибольшему натуральному числу, которое не превосходит  $X$  и содержит при этом ровно три единицы в своей двоичной записи. Если такого числа нет, то  $\alpha(X) = -1$ . Алиса захотела проверить свою гипотезу, но так как ей надо проверить 2023 посылки, придется вам написать программу, которая по заданному  $X$  считает  $\alpha(X)$ . Помогите Алисе! (Иначе получите реджект)

### Формат входных данных

В первой строке содержится единственное число  $1 \leq T \leq 10^5$  — количество тестовых кейсов. В  $i$ -й из следующих  $T$  строк содержится  $i$ -й тест, заданный единственным натуральным числом  $1 \leq X_i \leq 10^{18}$ .

### Формат выходных данных

Для каждого тестового кейса выведите в отдельной строке ответ на него —  $\alpha(X_i)$ .

### Пример

стандартный ввод	стандартный вывод
4	-1
1	7
8	896
1023	7
7	

### Замечание

В первом примере даны 4 тест-кейса.

- $X = 1$ . Положительного числа меньше 1 с тремя единицами в двоичной записи не существует.
- $X = 8$ . Тут нам подходит 7, так как двоичная запись этого числа 111.
- $X = 1023$ . Заметим, что число  $1023 = 1024 - 1$ , то есть оно имеет 10 единиц в своей записи, оставим из них старшие 3 и получим  $1110000000_2 = 896$
- $X = 7$ . В этом числе уже 3 бита, поэтому ответ будет 7.



## [C-В'] 3. Яблоки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вы любите яблоки и за сегодня хотите съесть  $x$  зеленых яблок и  $y$  красных.

У вас в саду поспело уже  $a$  зеленых яблок сладостью  $p_1, p_2, \dots, p_a$  и  $b$  красных яблок сладостью  $q_1, q_2, \dots, q_b$ . А еще у вас растет  $c$  магических яблок сладостью  $r_1, r_2, \dots, r_c$ , которые вы умеете превращать перед едой в красные или зеленые.

Понятное дело, что вы хотите съесть яблоки с наибольшей суммарной сладостью.

Найдите максимально возможную сумму сладости съеденных яблок, которая может быть достигнута при оптимальном превращении магических яблок.

### Формат входных данных

В первой строке записано пять чисел  $x, y, a, b, c$  ( $1 \leq x \leq a \leq 10^5, 1 \leq y \leq b \leq 10^5, 1 \leq c \leq 10^5$ ).

Во второй строке записано  $a$  чисел  $p_i$  ( $1 \leq p_i \leq 10^9$ ) — сладость зеленых яблок

В третьей строке записано  $b$  чисел  $q_i$  ( $1 \leq q_i \leq 10^9$ ) — сладость красных яблок

В четвертой строке записано  $c$  чисел  $r_i$  ( $1 \leq r_i \leq 10^9$ ) — сладость магических яблок

### Формат выходных данных

Выведите максимально возможную сумму сладости съеденных яблок.

### Примеры

стандартный ввод	стандартный вывод
1 3 2 3 4 5 6 1 2 3 10 11 12 13	46
1 1 2 2 1 10 4 5 6 1	16

## [C-B'] 4. Жизнь в казарме

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Учеба в Высшей Школе Психологии не понравилась Саше настолько, что он предпочел ей армию. И не зря! Всего за полгода службы он дорос до командира отряда.

В его рутинную обязанность входит каждое утро выстраивать своих подчинённых в определенном порядке. Однако, они, в отличие от него, ничего не смыслят в перестановках, а потому этим утром выстроились как попало.

Всего у Саши  $n$  подчиненных, пронумерованных от 1 до  $n$ , что очень удобно, потому что ему не приходится запоминать их по именам. Этим утром на  $i$ -е слева место встал подчиненный  $a_i$ , хотя там всегда стоял подчиненный  $b_i$ !

У Саши есть возможность исправить ситуацию, неограниченное количество раз отдав приказ следующего вида: «Подчиненный  $x$ , переместись на  $y$  позиций влево!», где  $y$  не больше, чем текущее количество человек слева от  $x$ .

Утро не резиновое, поэтому Саша хочет достичь результата, отдав как можно меньше приказов.

### Формат входных данных

Первая строка содержит одно целое число  $n$  ( $1 \leq n \leq 10^5$ ).

Вторая строка содержит  $n$  различных целых чисел  $a_1, \dots, a_n$  ( $1 \leq a_i \leq n$ ).

Третья строка содержит  $n$  различных целых чисел  $b_1, \dots, b_n$  ( $1 \leq b_i \leq n$ ).

### Формат выходных данных

Выведите в единственной строке минимальное количество требуемых приказов.

### Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 1 2 3 4	0
5 1 2 3 4 5 5 1 4 2 3	2

### Замечание

В первом примере никаких приказов отдавать не приходится — порядок уже правильный.

Во втором примере можно сначала переставить 4 на 2 позиции влево, а затем 5 на 4 позиции влево. Можно показать, что за меньшее количество приказов сделать правильный порядок не получится.

## [C-B'] 5. Экономическая задача

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В вашей любимой игре вышло новое обновление, в котором добавили много всего нового.

Ваши любимые разработчики добавили  $n$  предметов, но, так как они тоже люди и им тоже хочется кушать, бесплатно вы эти предметы не получите. Цена  $i$ -го предмета равна  $c_i$  плюгенов. При этом разработчики понимают, что если сделать цены слишком большими (а они так и сделали), то игроки могут разозлиться и пойти играть в какую-нибудь другую игру. Поэтому эти добрые люди добавили  $m$  купонов,  $i$ -й из которых позволяет уменьшить стоимость ровно одного предмета на  $d_i$  в том случае, если изначальная цена была хотя бы  $l_i$  плюгенов. Но важно не забывать, что каждый купон может быть использован **не более одного раза**. Более того, чтобы игроки не забывали, во что они играют, им запрещено применять на один предмет более одного купона.

Вы — простой студент одной известной экономической школы, поэтому перед вами встала очень сложная задача — потратить как можно меньше плюгенов для покупки всех  $n$  предметов (потому что по отдельности эти предметы никакого интереса не представляют).

### Формат входных данных

В первой строке даны целые числа  $n$  и  $m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — количество предметов и купонов.

Во второй строке перечислены числа  $c_1, \dots, c_n$  ( $1 \leq c_i \leq 10^9$ ) — стоимости предметов.

В третьей строке перечислены числа  $l_1, \dots, l_m$  ( $1 \leq l_i \leq 10^9$ ) — минимальные цены предметов для использования купонов.

В четвертой строке перечислены числа  $d_1, \dots, d_m$  ( $1 \leq d_i \leq l_i$ ) — скидки после использования купонов.

### Формат выходных данных

В единственной строке выведите минимальное количество плюгенов, которое потребуется для покупки всех предметов.

### Примеры

стандартный ввод	стандартный вывод
3 3 4 3 1 4 4 2 2 3 1	4
10 5 9 7 1 5 2 2 5 5 7 6 7 2 7 8 2 3 2 4 1 2	37

### Замечание

Рассмотрим первый пример. Если использовать второй купон (он уменьшает на 3 стоимость предмета, который стоил хотя бы 4 плюгена) для первого предмета и третий купон (он уменьшает на 1 стоимость предмета, который стоил хотя бы 2 плюгена) для второго предмета, то потребуется  $(4 - 3) + (3 - 1) + 1 = 4$  плюгена.

## [C-V] 6. Приготовление капкейков

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ваша знакомая Юлия приготовила  $n$  вкусных капкейков. Каждый из них обладает тремя параметрами — «Вкус», «Количество посыпки», «Количество крема».  $i$ -й капкейк имеет вкус  $t_i$ ,  $d_i$  посыпки и  $c_i$  крема сверху. Юлия не использует метрическую систему, поэтому эти числа могут быть равны нулю или даже быть отрицательными.

Для кружка, в котором она ведет занятия, ей нужно взять ровно  $m$  капкейков. Пусть *качество набора* это  $|\text{суммарный вкус}| + |\text{суммарное количество посыпки}| + |\text{суммарное количество крема}|$ , где  $|x|$  означает модуль числа  $x$ . Иначе говоря, если выбраны капкейки  $i_1, \dots, i_m$ , то *качество набора* равно  $|t_{i_1} + t_{i_2} + \dots + t_{i_m}| + |d_{i_1} + d_{i_2} + \dots + d_{i_m}| + |c_{i_1} + c_{i_2} + \dots + c_{i_m}|$ . При этом, так как Юлия сделала по одному пирожному каждого вида, каждый капкейк можно взять не более одного раза.

Девушка устала после такой масштабной готовки, поэтому теперь вам придется помочь ей выбрать пирожные так, чтобы максимизировать *качество набора*.

### Формат входных данных

В первой строке даны целые числа  $n$  и  $m$  ( $1 \leq n \leq 1000$ ,  $0 \leq m \leq n$ ) — количество капкейков, приготовленных Юлией, а также количество капкейков, необходимое для набора.

В следующих  $n$  строках перечислены тройки чисел  $t_i, d_i, c_i$  ( $-10^{10} \leq t_i, d_i, c_i \leq 10^{10}$ ) — параметры  $i$ -го капкейка.

### Формат выходных данных

В единственной строке выведите максимально возможное значение *качества набора*.

## Примеры

стандартный ввод	стандартный вывод
5 3 3 1 4 1 5 9 2 6 5 3 5 8 9 7 9	56
5 3 1 -2 3 -4 5 -6 7 -8 -9 -10 11 -12 13 -14 15	54
10 5 10 -80 21 23 8 38 -94 28 11 -26 -2 18 -69 72 79 -26 -86 -54 -72 -50 59 21 65 -32 40 -94 87 -62 18 82	638
3 2 2000000000 -9000000000 4000000000 7000000000 -5000000000 3000000000 6000000000 -10000000000 8000000000	30000000000

## Замечание

В первом примере надо взять капкейки с номерами 2, 4 и 5. Тогда *качество набора* будет равно  $|1 + 3 + 9| + |5 + 5 + 7| + |9 + 8 + 9| = 13 + 17 + 26 = 56$ .

Во втором примере надо взять пирожные с номерами 1, 3 и 5. Тогда *качество набора* будет равно  $|1 + 7 + 13| + |(-2) + (-8) + (-14)| + |3 + (-9) + 15| = 21 + 24 + 9 = 54$ .

В третьем примере надо взять капкейки с номерами 3, 4, 5, 7, 10. Тогда *качество набора* будет равно  $|(-323)| + |66| + |249| = 638$ .

## [В'-В] 1. Жизнь в общежитии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Егор поступил в Высшую Школу Психологии на программу (сюрприз-сюрприз) психологии! Его поселили в коридорное общежитие, где он решил провести свое первое психологическое исследование.

Всех первокурсников поселили на один этаж, на котором до этого никто не жил. Этаж представляет из себя прямую, а комнаты — точки на этой прямой. В этом году в общежитие поселились  $n$  первокурсников, пронумерованных от 1 до  $n$ ,  $i$ -го из которых заселили в комнату на позиции  $x_i$  (все позиции комнат различные).

По прошествии нескольких дней Егор заметил, что некоторые из его ранее необщительных друзей, которые вместе с ним поселились в общежитие, внезапно становились экстравертами. Понаблюдав за ситуацией дальше, он смог понять как именно и на кого повлияла жизнь в общежитии.

Каждый первокурсник к моменту заселения был либо интровертом, либо экстравертом. Егор выяснил, что существует число  $R$  — так называемый коэффициент общительности (значение этого коэффициента Егор вычислить не смог). Для фиксированного  $R$  выполняется следующий процесс:

1. Если существует такая пара студентов  $i$  и  $j$  ( $i \neq j$ ), что  $i$  интроверт, а  $j$  — экстраверт, а также расстояние между комнатами  $i$  и  $j$  не больше  $R$  (т.е.  $|x_i - x_j| \leq R$ ), то процесс продолжается дальше. Иначе считается, что настроение в общежитии стабилизировалось и процесс прекращается.
2. Студенты  $i$  и  $j$  начинают общаться и по вечерам вместе пить милкшейки. Спустя некоторое время студент  $i$  становится экстравертом.
3. Процесс продолжается дальше, начиная с шага 1.

Егор быстро понял, что настроение в общежитии уже стабилизировалось, и для каждого студента, используя психологические тесты из интернета, выяснил является он интровертом или экстравертом. Теперь ему интересно какое минимальное количество экстравертов могло поселиться в общежитие в начале этого года.

### Формат входных данных

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 1000$ ).

Каждая из следующих  $n$  строк содержит два целых числа  $x_i$  ( $0 \leq x_i \leq 10^6$ ) и  $s_i$ , где  $s_i = 0$ , если  $i$ -й студент — интроверт, и  $s_i = 1$  иначе.

### Формат выходных данных

В единственной строке выведите минимальное количество экстравертов, поселившихся в общежитии в начале этого года.

### Пример

стандартный ввод	стандартный вывод
6 1 1 10 0 6 1 3 1 15 1 7 1	3

## [B'-B] 2. Перекладывание JSONов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Наступило лето, и первокурсник Джейсон наконец-то отправился стажироваться! Придя в офис в свой первый рабочий день, он обнаружил, что в помещении находится  $n$  рядов по  $m$  столов в каждом. На каждом из столов лежит сколько-то JSONов. Директор компании, в которой стажирется Джейсон, очень любит четные числа. В связи с этим Джейсону было поручено переложить JSONы таким образом, чтобы максимизировать количество столов, на которых лежит четное количество JSONов.

Скажем, что  $j$ -й стол в  $i$ -м ряду имеет координаты  $(i, j)$ .

Джейсону известно, что на столе с координатами  $(i, j)$  лежит  $a_{ij}$  JSONов. За одну операцию он может выбрать произвольный стол с координатами  $(i, j)$ , на котором есть хотя бы один JSON, и переложить ровно один JSON на один из столов с координатами  $(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$ ,  $(i, j + 1)$  (разумеется, стол с такими координатами должен существовать). Силы Джейсона не бесконечны, поэтому он может сделать не более  $n \cdot m$  таких операций.

### Формат входных данных

Первая строка ввода содержит числа  $n$  и  $m$  ( $1 \leq n, m \leq 500$ ) — количество рядов и количество столов в одном ряду соответственно.

Следующие  $n$  строк содержат описание офиса.  $i$ -я из этих строк содержит записанные через пробел числа  $a_{i1}, a_{i2}, \dots, a_{im}$ , где  $a_{ij}$  ( $0 \leq a_{ij} \leq 9$ ) — количество JSONов на столе с координатами  $(i, j)$ .

### Формат выходных данных

В первой строке выведите одно число от 0 до  $n \cdot m$  — количество выполненных операций.

Далее выведите описания операций — по одному описанию в одной строке.

Описание операции должно представлять из себя четыре числа  $x_1, y_1, x_2, y_2$ , записанные через пробел, где  $(x_1, y_1)$  — координаты стола, с которого Джейсон взял JSON, а  $(x_2, y_2)$  — координаты стола, на который Джейсон положил JSON. На столе  $(x_1, y_1)$  на момент начала выполнения операции должен быть хотя бы один JSON, и клетки  $(x_1, y_1)$  и  $(x_2, y_2)$  должны быть соседними по стороне.

Если ответов несколько, выведите любой из них. Обратите внимание, что не нужно минимизировать количество операций.

### Примеры

стандартный ввод	стандартный вывод
2 3 5 7 0 1 2 1	3 1 1 1 2 2 3 2 2 2 2 2 1
2 3 0 1 0 1 2 1	2 1 2 1 3 1 3 2 3
1 7 7 7 7 7 7 7	3 1 1 1 2 1 3 1 4 1 5 1 6

## [В'-В] 3. Перестановки и счастье

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Перестановкой назовем массив, состоящий из  $n$  различных целых чисел от 0 до  $n - 1$  в произвольном порядке. Например,  $[2, 1, 0, 4, 3]$  является перестановкой, а  $[0, 1, 1]$  и  $[0, 2]$  — нет.

На день рождения Пете подарили перестановку  $p_1, p_2, \dots, p_n$ . Счастье Пети определяется как количество пар  $(i, j)$ , таких что  $1 \leq i < j \leq n$  и  $p_i > p_j$ .

Пете разрешено многократно выполнять следующую операцию: переместить первый элемент перестановки в ее конец. Например, если изначально у Пети была перестановка  $[1, 2, 0, 4, 3]$ , то после применения операции у него будет перестановка  $[2, 0, 4, 3, 1]$ .

Для каждого  $k$  от 0 до  $n - 1$  выведите значение счастья Пети, если он выполнит данную операцию  $k$  раз.

### Формат входных данных

Первая строка содержит число  $n$  ( $2 \leq n \leq 3 \cdot 10^5$ ) — размер перестановки.

Вторая строка содержит записанные через пробел числа  $p_1, p_2, \dots, p_n$  — перестановку чисел от 0 до  $n - 1$ .

### Формат выходных данных

В следующих  $n$  строках выведите числа  $a_0, a_1, \dots, a_{n-1}$ , где  $a_k$  — счастье Пети при условии, что он  $k$  раз переместил первый элемент перестановки в ее конец.

### Примеры

стандартный ввод	стандартный вывод
5 0 1 2 3 4	0 4 6 6 4
10 0 8 1 2 7 4 5 3 6 9	13 22 15 22 27 22 23 22 25 22
5 0 4 1 3 2	4 8 4 6 4



## [В'-В] 4. Раздача в столовой

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В столовой одного известного Завода вот-вот начнётся обед. На обеде есть три гарнира — макарошки, пюрешка и греча. К гарнирам, конечно же, полагается горячее. Горячих блюд тоже три (ну и совпадение!) — сосиски, котлетки и кура. На раздаче сегодня стоит добрая повариха Алевтина Мстиславовна. Она немного ленива, но очень любит заводчан, вот перед ней и встала задача.

Итак, для начала поведаем вам прописные истины: макарошки лучше сочетать с сосисками, пюрешку с котлетками, а гречу с курой. Человек, у которого на обед одно из этих трёх сочетаний, будет счастливым, в противном случае счастливым он не будет.

На раздаче в очередь выстроились  $n$  заводчан. По очереди они будут подходить к Алевтине и называть ей гарнир, который они хотят. У Алевтины на раздаче будут все три гарнира, а также одно из горячих блюд. Заводчанину она будет выдавать гарнир, который он хочет, а также то горячее которое сейчас на раздаче. Кроме того, за всё время Алевтина готова не более чем  $k$  раз сходить в Горячий Цех и заменить горячее блюдо на раздаче на любое другое. Естественно, в начале она вольна принести на раздачу любое горячее блюдо, которое захочется.

Алевтина дружит со всеми заводчанами и уже заранее знает кому какой гарнир хочется. Всей информацией она поделилась с Вами. Алевтина добра душой и хочет, чтобы как можно больше заводчан были счастливы благодаря сочетанию гарнира и горячего. Ваша задача — сказать какое наибольшее число заводчан могут оказаться счастливыми при заданном ограничении на число замен горячего блюда на раздаче. Не подведите Алевтину!

### Формат входных данных

В первой строке входных даны целые числа  $n$  и  $k$  ( $1 \leq n \leq 10^5$ ,  $0 \leq k \leq 20$ ). В следующих  $n$  строках даны предпочтения по гарнирам в том порядке, в котором заводчане стоят в очереди, по одному на строке. Предпочтение описано одним из трёх символов: «М», «Р» или «G» — макарошки, пюрешка или гречка соответственно.

### Формат выходных данных

Выведите какое наибольшее число заводчан будут счастливыми после обеда.

### Пример

стандартный ввод	стандартный вывод
5 1 М М G М Р	4

### Замечание

Запасы гарниров и горячих блюд на раздаче считайте неограниченными.

Разберём пример. В примере последовательность желаемых гарниров такая: макарошки, макарошки, греча, макарошки, пюре; Алевтина 1 раз может поменять горячее на раздаче. Возможный вариант того, как ей поступить, следующий: сначала принести сосиски, раздать их 4-ём людям (соответственно первый, второй и четвёртый человек будут счастливы), а затем заменить горячее на котлетки (и тогда пятый человек тоже будет счастлив). Таким образом Алевтина осчастливит 4 заводчан.

## [В'-В] 5. Кабанчик арбузер

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Есть такой уважаемый кабанчик — Бузизнесмен Пиджакович, для друзей коротко Бузя. И он, увы, самый настоящий арбузер. Своих друзей он делает несчастными, когда те счастливы, а когда те несчастны — наоборот, пытается сделать их счастливыми. Арбузы тут тоже неспроста — друг Бузи счастлив тогда и только тогда, когда у него есть арбуз!

Бузя — кабанчик на подскоке. А именно, у Бузи может быть одно натуральное число  $k$  — сила подскока, его он выберет самостоятельно.

Друзья Бузи выстроились в шеренгу один за другим. Про каждого известно есть у него арбуз или нет. Когда он выберет силу подскока, он будет надоедать своим друзьям. А именно, он может делать следующее действие столько раз, сколько захочет:

1. Встать рядом с любым из своих друзей
2. Подскокить и пробежать прямо (не сворачивая и не разворачиваясь; по направлению к правому концу шеренги) вдоль не менее чем  $k$  своих друзей (напоминаем:  $k$  это заранее выбранное число — сила подскока). Естественно, он не может этого сделать, если с начального места далее в шеренге меньше чем  $k$  друзей (более подробно посмотрите в примечании).
3. Пробегая вдоль каждого из своих друзей, Бузя посмотрит — был ли у друга арбуз или нет. Если был — Бузя его разобьет, и у друга арбуза больше не будет. Если же в данный момент арбуза у друга не было — Бузя даст ему арбуз, таким образом у друга теперь будет арбуз.

Бузя конечно проблемный, но хочет, чтобы у всех его друзей в итоге были арбузы. При этом он хочет выбрать наибольшую силу подскока. Но, конечно же, сила подскока не должна превышать количества друзей у Бузи (кабанчик хоть и уважаемый, но не выскочка).

### Формат входных данных

Вам дана одна строка  $s$  ( $1 \leq |s| \leq 10^5$ ,  $|s|$  — это длина строки), описывающая наличие арбузов у друзей Бузи, в том порядке в котором они стоят в шеренге. Строка состоит из символов «N» и «A». «N» значит, что у очередного друга Бузи нет арбуза, а «A» — что арбуз есть (это сокращения от словосочетаний «Net arbuza» и «Arbuz», соответственно).

### Формат выходных данных

Выведите одно натуральное число  $k \leq |s|$  — наибольшую силу подскока, которую может выбрать Бузя, так, чтобы выполняя описанные в условии действия у всех друзей Бузи оказались арбузы.

### Примеры

стандартный ввод	стандартный вывод
ANA	2
NAAAAAAAA	8
AAAANNNN	4

### Замечание

Приведём пример того, какую операцию Бузя не может выполнить: если всего друзей 5 (про-нумеруем их с 1), Бузя выбрал силу подскока  $k = 3$ , а затем встал у 4-го друга, то он не может пробежать вдоль трёх друзей по направлению к концу шеренги, так как всего по пути 2 друга: четвёртый и пятый. Аналогично он не может пробежать от 5-го друга прямо при  $k = 3$ , так как друг всего один.

Разберём первый пример. Изначально наличие арбузов у друзей следующее: есть, нету, есть. Далее, выбрав  $k = 2$ , можно сделать следующее. Подскакиваем от первого друга и пробегаем вдоль

двоих, после этого состояние: нету, есть есть. Далее можно подскочить от второго друга и пробежать вдоль двоих, после этого состояние: нету, нету, нету. Наконец, подскакиваем около первого друга и бежим вдоль всех трёх, после чего состояние: есть, есть, есть. Да, последнее действие корректно, так как пробежать надо вдоль **не менее** чем  $k$  друзей ( $a \geq 2$ ).

Запасы арбузов и выносливости у Бузи неограниченны. Он же кабанчик.

## [В'-В] 6. Расселение мигрантов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

Недавно Центр Помощи Мигрантам получил в свое распоряжение новое общежитие, которое представляет собой большой коридор, в котором последовательно расположены  $m$  комнат. Так как общежитие является достаточно большим, администрация Центра Помощи Мигрантам решила, что в каждой комнате будет проживать не более одного мигранта.

Для удобства пронумеруем все комнаты общежития целыми числами от 1 до  $m$ . Назовем комнаты  $i$  и  $j$  *соседними*, если  $|i - j| = 1$ .

Уже давно в очереди на получение общежития находится  $n$  мигрантов. По счастливому стечению обстоятельств  $n \leq m$ , то есть в новом общежитии хватит мест, чтобы разместить всех мигрантов, находящихся в очереди. По результатам опроса для каждого мигранта были вычислены два параметра:  $a_i$  и  $b_i$ .

Параметр  $a_i$  характеризует уровень счастья  $i$ -го мигранта при условии, что хотя бы в одной из соседних с ним комнат будет проживать другой мигрант. Параметр  $b_i$  характеризует уровень счастья  $i$ -го мигранта при условии, что во всех соседних с ним комнатах не будут проживать другие мигранты.

Администрация Центра Помощи Мигрантам поставила для себя задачу — расселить всех мигрантов таким образом, чтобы максимизировать суммарный уровень их счастья от проживания в общежитии. Помогите этого достичь и вычислите максимально возможный суммарный уровень счастья мигрантов.

### Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число  $t$  ( $1 \leq t \leq 100\,000$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка описания набора входных данных содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 500\,000$ ,  $1 \leq m \leq 10^9$ ,  $n \leq m$ ) — количество мигрантов в очереди, а также количество комнат в общежитии.

Каждая из следующих  $n$  строк описания набора входных данных содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq 10^9$ ) — уровень счастья  $i$ -го мигранта при условии наличия соседей и при условии отсутствия соседей, соответственно.

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $10^6$ .

### Формат выходных данных

Для каждого набора входных данных выведите одно целое число — максимально возможный суммарный уровень счастья всех мигрантов.

### Пример

стандартный ввод	стандартный вывод
3	50
1 100	40
100 50	40
2 100	
10 20	
30 10	
4 5	
1 10	
10 1	
10 1	
10 1	

## Замечание

В первом примере в очереди находится всего один мигрант. Поэтому при любом расселении у него не будет соседей, а значит уровень его счастья будет равен 50.

Во втором примере в очереди находятся два мигранта. Если их поселить в соседние комнаты, суммарный уровень их счастья будет равен  $10 + 30 = 40$ . Если их поселить не в соседние комнаты, суммарный уровень их счастья будет равен  $20 + 10 = 30$ .

В третьем примере можно, например, поселить первого мигранта в комнату 1, а остальных трех мигрантов — в комнаты 3, 4 и 5.

## [В-А'] 1 Петя и бинарные числа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Петя — фанат brawl stars, однако, с недавнего времени он не может играть в свою любимую игру. Очевидно, что вместо brawl stars Петя увлекся бинарными числами.

Петя считает, что красота числа  $x$  равна количеству единиц в бинарном представлении числа  $x$ . Например, красота 0 равна 0, красота 1 равна 1, а красота 5 равна 2.

Поскольку у Пети осталась жилка игрока в brawl stars, то он любит делать какие-то странные действия, а именно за одну операцию из числа  $x$  Петя получает число, равное остатку от деления  $x$  на красоту числа  $x$ .

Петя считает, что великолепность числа  $x$  равна количеству таких операций, которые нужно сделать, чтобы получить 0. Можно доказать, что великолепность любого целого числа всегда является конечным числом. Например, великолепность 0 равна 0, а великолепность 5 равна 2.

Петя выиграл в доту бинарное число  $x$ , состоящее из  $n$  цифр. Для всех целых  $i$ , таких что  $1 \leq i \leq n$ ,  $x_i$  равно числу  $x$  с инвертированным  $i$ -м битом слева.

Поскольку Петя увлечен дотой больше, чем бинарными числами, он просит вас посчитать великолепность чисел  $x_1, x_2, \dots, x_n$ .

### Формат входных данных

В первой строке находится одно натуральное число  $n$  — количество цифр в бинарном числе  $x$  ( $1 \leq n \leq 2 \times 10^5$ ).

Во второй строке находится бинарное число  $x$ , состоящее из  $n$  цифр. Каждая цифра равна единице или нулю. Обратите внимание, что  $x$  может содержать ведущие нули.

### Формат выходных данных

Выведите  $n$  строк,  $i$ -я из которых содержит одно число, равное великолепности  $x_i$ .

### Примеры

стандартный ввод	стандартный вывод
5	2
01101	2
	2
	3
	1
1	1
0	

### Замечание

Пояснение к первому примеру:

$x_1 = 29$ . Красота 29 равна 4, значит после первой операции Петя получит  $29 \bmod 4 = 1$ . Красота 1 равна 1, значит после второй операции он получит  $1 \bmod 1 = 0$ , то есть великолепность  $x_1$  равна 2.

$x_2 = 5$ . Красота 5 равна 2, значит после первой операции Петя получит  $5 \bmod 2 = 1$ . Красота 1 равна 1, значит после второй операции он получит  $1 \bmod 1 = 0$ , то есть великолепность  $x_2$  равна 2.

$x_3 = 9$ . Красота 9 равна 2, значит после первой операции Петя получит  $9 \bmod 2 = 1$ . Красота 1 равна 1, значит после второй операции он получит  $1 \bmod 1 = 0$ , то есть великолепность  $x_3$  равна 2.

$x_4 = 15$ . Красота 15 равна 4, значит после первой операции Петя получит  $15 \bmod 4 = 3$ . Красота 3 равна 2, значит после второй операции он получит  $3 \bmod 2 = 1$ . Красота 1 равна 1, значит после третьей операции он получит  $1 \bmod 1 = 0$ , то есть великолепность  $x_4$  равна 3.

$x_5 = 12$ . Красота 12 равна 2, значит после первой операции Петя получит  $12 \bmod 2 = 0$ . То есть великолепность  $x_5$  равна 1.

## [B-A'] 2. Шоу талантов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В сегодняшнем шоу талантов  $n$  участников, после их выступлений они уже получили оценки за технику исполнения.  $i$ -й участник получил  $a_i$  очков. Теперь их судьба находится в руках  $m$  зрителей. Каждый зритель должен выбрать ровно  $v$  различных участников, их счет при этом увеличится на 1.

После того, как все голоса учтены, участники сортируются по убыванию их суммарного счета и первые  $p$  считаются победителями. Если несколько участников набирают одинаковый счет, то их порядок определяется организатором шоу.

Всем безумно интересно, кто же может оказаться среди победителей по результатам голосования. Посчитайте сколько из данных  $n$  участников могут оказаться в победителях.

### Формат входных данных

В первой строке входных данных даны четыре целых числа  $n, m, v, p$ . ( $2 \leq n \leq 10^5, 1 \leq m \leq 10^9, 1 \leq v, p \leq n - 1$ ) - количество участников, количество зрителей, количество голосов у каждого зрителя и количество победителей в шоу.

Во второй строке даны  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) - изначальное количество очков у каждого участника.

### Формат выходных данных

Выведите единственное число - количество участников, которые могут оказаться среди победителей по результатам голосования.

### Примеры

стандартный ввод	стандартный вывод
6 1 2 2 2 1 1 3 0 2	5
6 1 5 2 2 1 1 3 0 2	3
10 4 8 5 7 2 3 6 1 6 5 4 6 5	8

## [В-А'] 3. Двухцветный бильярд

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ваня придумал свои правила для бильярда. В его игре участвует  $n$  белых и  $n$  черных шаров, шары каждого цвета пронумерованы целыми числами от 1 до  $n$ .

Шары для игры укладываются в длинную коробочку в один ряд. После игры Ваня сложил их и получилась последовательность номеров шаров  $p_1, p_2, \dots, p_{2n}$  слева направо и последовательность цветов шаров  $t_1, t_2, \dots, t_{2n}$  слева направо.

Чтобы было удобно выкладывать шары из коробочки для следующей игры, Ваня хочет, чтобы белые и черные шары располагались в порядке  $1, 2, \dots, n$  слева направо. При этом цвета могут быть перемешаны между собой произвольным образом, важен только порядок среди шаров одинакового цвета.

Чтобы достичь такого расположения, Ваня может менять местами любые два соседних шара. Какое минимальное количество операций обмена соседних шаров нужно будет совершить, чтобы получить хорошую последовательность шаров?

### Формат входных данных

В первой строке находится единственное целое число  $n$  ( $1 \leq n \leq 2000$ ) — количество шаров каждого цвета.

В следующих  $2n$  строках находится описание изначального порядка шаров, в  $i$ -й строке находится символ  $t_i$  и целое число  $p_i$  ( $t_i \in \{B, W\}$ ,  $1 \leq p_i \leq n$ ) — цвет и номер  $i$ -го шара.  $t_i = B$  соответствует черному цвету,  $t_i = W$  соответствует белому цвету.

Гарантируется, что среди цветов  $t_1, t_2, \dots, t_{2n}$  есть ровно  $n$  белых и ровно  $n$  черных цветов.

Гарантируется, что среди номеров  $p_1, p_2, \dots, p_{2n}$  каждый номер от 1 до  $n$  встречается ровно два раза.

### Формат выходных данных

Выведите единственное целое число — минимальное количество операций обмена соседних шаров, которое нужно сделать, чтобы получить последовательность шаров, в которой белые и черные шары располагаются в порядке  $1, 2, \dots, n$  слева направо.



## Примеры

стандартный ввод	стандартный вывод
2 W 1 B 2 W 2 B 1	2
3 W 2 B 1 W 1 B 3 W 3 B 2	4
4 B 3 W 3 W 4 B 1 B 4 B 2 W 2 W 1	14

## Замечание

В первом тесте изначальная последовательность шаров:  $[1, \underline{2}, 2, \underline{1}]$ . Номера черных шаров подчеркнуты.

Можно сделать следующие операции:

- Обмен третьего и четвертого шаров. После операции последовательность шаров:  $[1, \underline{2}, \underline{1}, 2]$ .
- Обмен второго и третьего шаров. После операции последовательность шаров:  $[1, \underline{1}, \underline{2}, 2]$ .

Полученная после этих двух операций последовательность шаров будет хорошей.

Можно показать, что с помощью меньше чем двух операций получить хорошую последовательность шаров невозможно.

## [В-А'] 4. Побег Гориллы Коко

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

После того, как Горилла Коко в очередной раз сбежала из секретной московской лаборатории по изучению приматов и навела немало шумихи и беспредела в окрестностях столицы, ее наконец поймали и вернули в лабораторию для проведения новых экспериментов.

Для того, чтобы горилла в очередной раз не сбежала, ее поместили в большой вольер прямоугольной формы, имеющий размер  $n \times m$ , а в каждом углу вольера разместили сотрудника лаборатории с электрошокером в руках. Таким образом, шансов сбежать у Коко на этот раз практически нет. Однако, горилла в очередной раз доказала всем, что она не настолько глупа.

Для удобства представим себе вольер как прямоугольник на плоскости. Границы прямоугольника являются стенами вольера. Его углы расположены в точках с координатами  $(0, 0)$ ,  $(n, 0)$ ,  $(n, m)$  и  $(0, m)$  — именно в этих точках расположены сотрудники лаборатории.

Оказавшись в вольере, горилла обнаружила коричневый сгнивший кокос, издававший весьма противный характерный запах. Изначально кокос лежал в точке с координатами  $(x_1, y_1)$ . Немного погуляв по вольеру, горилла также обнаружила немалых размеров камень, расположенный в точке с координатами  $(x_2, y_2)$ . В голове гориллы тут же сложился план побега.

Коко решила действовать следующим образом. Сначала она возьмет в лапы кокос и со всей силы бросит его в выбранном ей направлении. Кокос будет лететь в заданном направлении, отражаясь от стен вольера по классическим законам отражения, известным из физики (угол, с которым кокос подлетит к стене, будет равен углу, с которым он отскочит от стены). Если в какой-то момент времени кокос прилетит ровно в точку, в которой лежит камень, он тут же остановится и упадет на землю в той же точке, а сам камень начнет двигаться в том же направлении, куда летел кокос в момент удара. Камень будет двигаться и отражаться от стен по тем же законам, что и кокос. Если вы удивляетесь и думаете, что это физически невозможно, то вы недооцениваете силу и рвение на волю Гориллы Коко!

Цель Коко — бросить кокос таким образом, чтобы он в некоторый момент времени ударился о камень таким образом, чтобы тот в некоторый момент времени ударил по голове и оглушил какого-либо сотрудника лаборатории (иными словами, камень должен попасть в точку, являющуюся углом прямоугольника).

К сожалению, все не так просто, и есть несколько ограничений.

Во-первых, камень во время своего полета не должен сталкиваться с кокосом, потому что если это произойдет, кокос лопнет от силы удара, его «аромат» разойдется по всему вольеру, после чего сотрудники лаборатории разозлятся и ударят гориллу электрошокером.

Во-вторых, суммарно кокос и камень должны удариться о стены вольера не более, чем  $k$  раз, потому что каждый удар создает лишний шум и злит сотрудников лаборатории.

Наконец, горилла хочет бросить кокос таким образом, чтобы минимизировать суммарное количество ударов кокоса и камня о стены вольера.

Горилла Коко была достаточно умна, чтобы придумать данный план, однако на его реализацию умственных способностей ей все же не хватило. Помогите горилле вычислить минимально возможное суммарное количество ударов кокоса и камня о стены вольера, соблюдая условия, описанные выше.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $3 \leq n, m \leq 1000$ ) — размеры вольера.

Вторая строка содержит одно целое число  $k$  ( $0 \leq k \leq 200$ ) — максимально допустимое суммарное количество ударов кокоса и камня о стены вольера.

Третья строка содержит два целых числа  $x_1$  и  $y_1$  ( $1 \leq x_1 \leq n - 1$ ,  $1 \leq y_1 \leq m - 1$ ) — начальные координаты кокоса.

Четвертая строка содержит два целых числа  $x_2$  и  $y_2$  ( $1 \leq x_2 \leq n - 1$ ,  $1 \leq y_2 \leq m - 1$ ) — изначальные координаты камня.

Гарантируется, что изначальные положения кокоса и камня не совпадают.

### Формат выходных данных

Выведите одно целое число — минимально возможное суммарное количество ударов кокоса и камня о стены вольера до попадания камня в голову одного из сотрудников.

Если попасть камнем в голову, сделав не более  $k$  ударов о стены, невозможно, выведите число  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
5 4 200 1 2 3 2	2
5 4 200 3 2 1 2	1
5 4 200 1 1 2 3	-1

## [В-А'] 5. Задача про массивы

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В очередной скучный день Боб решил поиграться с массивами. Перед ним лежит массив  $a$  из  $n$  целых чисел с которым он может выполнять следующую операцию:

- Сначала Боб фиксирует целое положительное число  $k$ . После этого для каждого элемента последовательности  $a_i$  Боб либо заменяет  $a_i$  на  $a_i \bmod k$  либо ничего не делает. За такую операцию Боб платит  $2^k$  монет вне зависимости от количества измененных элементов массива.

Боб хочет узнать, может ли он получить из массива  $a$  массив  $b$  применив описанную выше операцию произвольное число раз. В случае, если это возможно, Боб также хотел бы узнать минимальную суммарную стоимость выполнения описанных операций.

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 50$ ) — размеры массивов.

Во второй строке содержатся  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 50$ ) — элементы массива  $a$ .

В третьей строке содержатся  $n$  целых чисел  $b_i$  ( $0 \leq b_i \leq 50$ ) — элементы желаемого массива  $b$ .

### Формат выходных данных

В единственной строке выходных данных выведите  $-1$ , если Боб не сможет получить массив  $b$ . В ином случае выведите минимальную стоимость операций.

### Примеры

стандартный ввод	стандартный вывод
4 34 23 5 49 0 0 0 0	2
4 4 5 3 15 4 5 2 15	-1
2 1 2 1 2	0
1 47 12	34359738368
1 50 2	8
2 50 47 2 14	8589934600

## [В-А'] 6. Майнкрафт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Боб фанат игры майнкрафт. Однажды играя на сервере он обнаружил, что у него совсем закончились руды.

Чтобы не терять время в шахте, Боб обратился за помощью к своим  $x + y + z$  друзьям.  $i$ -й друг предложил Бобу  $a_i$  меди,  $b_i$  железа и  $c_i$  золота. Чтобы не злоупотреблять помощью друзей Боб решил, что возьмет у  $x$  человек медь, у  $y$  человек железо и у  $z$  человек золото так, чтобы у каждого из людей Боб забрал ровно 1 вид руды.

Помогите Бобу и скажите, какое максимальное суммарное количество слитков руд он может получить, если оптимально попросит ресурсы у друзей.

### Формат входных данных

В первой строке входных данных содержатся 3 целых числа  $x, y, z$  ( $1 \leq x, y, z; x + y + z \leq 10^5$ ) — количество друзей у которых Боб попросит медь, железо и золото соответственно.

В следующих  $x + y + z$  строках содержатся тройки целых чисел  $a_i, b_i, c_i$  ( $1 \leq a_i, b_i, c_i \leq 10^9$ ) — количество меди, железа и золота, которое готов пожертвовать  $i$ -й друг.

### Формат выходных данных

В единственной строке выходных данных выведите ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
2 2 2 10 1 1 10 1 1 1 14 1 1 14 1 1 1 16 1 1 16	80
3 1 1 9 2 5 1 8 1 1 9 8 10 4 4 15 4 4	50
4 2 3 7 4 8 5 7 4 2 6 5 5 7 3 4 9 9 5 2 7 1 1 9 8 4 8 7 6 5	65

## [A'-A] 1. Практикум по физике

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На сегодняшнем практикуме по физике вы изучаете устойчивые положения равновесия. Учитель раздал каждому из вас по многоугольнику. У вашего многоугольника  $n$  вершин и его центр масс находится в точке  $(x_c, y_c)$ .

Положение многоугольника называется устойчивым, если две или более его точки стоят на столе (горизонтальной прямой), никакие другие точки не находятся ниже этой прямой и проекция центра масс многоугольника на прямую стола находится строго между какими-то двумя точками, стоящими на столе. Два положения равновесия считаются одинаковыми, если чтобы их достиг нужно повернуть многоугольник на один и тот же угол.

Ваша задача на этом практикуме — посчитать количество различных положений равновесия выданного вам многоугольника. Справитесь ли вы с такой задачей?

### Формат входных данных

В первой строке входных данных даны три целых числа  $n$ ,  $x_c$ ,  $y_c$  ( $3 \leq n \leq 1000$ ,  $-1000 \leq x_c, y_c \leq 1000$ ) — число вершин вашего многоугольника и координаты его центра масс. В каждой из следующих  $n$  строк дано по два целых числа  $x_i$ ,  $y_i$  ( $-1000 \leq x_i, y_i \leq 1000$ ) — координаты вершин многоугольника в порядке обхода против часовой стрелки. Гарантируется, что многоугольник не содержит самопересечений.

### Формат выходных данных

Выведите единственное число — количество устойчивых положений равновесия многоугольника.

### Пример

стандартный ввод	стандартный вывод
7 2 2 1 1 3 1 4 2 3 2 3 3 2 4 1 3	3

### Замечание

В первом тесте различные положения равновесия многоугольника задаются следующими парами точек, находящихся на столе:

1.  $(1, 3)$ ,  $(1, 1)$
2.  $(4, 2)$ ,  $(2, 4)$
3.  $(1, 1)$ ,  $(3, 1)$

## [A'-A] 2. Zombies are coming

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Во всем мире уже 4 года как идет зомби-апокалипсис. Вы сидите в своем домике в деревне на берегу реки, играете на балалайке песни для своего домашнего медведя, будучи полностью уверены, что вы в безопасности, но тут вы слышите странные крики. Вы выглядываете в окно, и видите плывущую на плотках армию зомби. Надо быстро что-то делать, вы выбегаете из дома и решаете, что надо как можно быстрее уезжать подальше по течению реки.

Для этого вы можете забежать в любой дом, находящийся на любом берегу реки, завести мотор у печки (самый быстрый и удобный вид транспорта в деревне), забрать из этого дома все поленья, которые являются топливом для печки, и выехать на ней из дома. После этого, вы можете сколько угодно раз (возможно 0) проехать по двусторонним мостам, проходящим от дома, у которого вы сейчас находитесь, до дома, находящегося на противоположном берегу, и забрать оттуда все поленья. Перемещаться по берегу реки нельзя, так как из леса в любой момент могут выпрыгнуть зомби. Также зомби двигаются по вашим следам, поэтому вам необходимо на каждом берегу реки перемещаться в дома, находящиеся все дальше и дальше по течению. То есть для каждого берега по отдельности номера посещенных домов должны возрастать. Какое максимальное количество поленьев вы можете собрать на своей печке, следуя данным правилам?

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $m$  и  $p$  ( $1 \leq n, m \leq 40\,000$ ,  $0 \leq p \leq 100\,000$ ) — количество домов на левом берегу, на правом берегу и количество мостов между ними соответственно.

В следующих  $n$  строках содержится по одному целому числу  $a_i$  ( $0 \leq a_i \leq 40\,000$ ) — количество поленьев в  $i$ -м доме на левом берегу реки.

В следующих  $m$  строках содержится по одному целому числу  $b_i$  ( $0 \leq b_i \leq 40\,000$ ) — количество поленьев в  $i$ -м доме на правом берегу реки.

В последних  $p$  строках содержится по два целых числа  $u$  и  $v$  ( $1 \leq u \leq n$ ,  $1 \leq v \leq m$ ) описывающие мост между домом  $u$  на левом берегу и  $v$  на правом. Между каждой парой домов существует не более одного моста.

### Формат выходных данных

Выведите одно целое число — максимальное количество поленьев, которые вы можете собрать.

### Пример

стандартный ввод	стандартный вывод
3 4 7 1 2 3 4 5 6 6 1 3 2 1 2 2 3 1 3 2 3 3 3 4	20

## Замечание

Есть два способа получить столько поленьев:  $1R - 2L - 2R - 3L - 3R$  и  $1R - 2L - 2R - 3L - 4R$  (L обозначают дома на левом берегу, R – на правом). Обратите внимание, что в первом способе нельзя вернуться в 1-й дом на левом берегу из 3-го на правом, так как это нарушает правило о том, что надо всегда двигаться дальше по течению.



## [A'-A] 3. Петя и странные функции

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Петю больше не интересуют бинарные числа, поэтому он начал изучать массивы. Его лучший друг Егор подарил ему массив  $a$ , состоящий из  $n$  целых чисел.

Назовем красотой подотрезка массива разность максимального и минимального числа на этом подотрезке.

Назовем привлекательностью разбиения массива на непустые подотрезки произведение красот этих подотрезков.

Разбиением массива на подотрезки называется разбиение на непересекающиеся множества подряд идущих элементов массива, дающие в объединении весь массив.

Когда Егор дарил массив Пете, у него было условие, что Петя должен посчитать сумму привлекательностей по всем возможным разбиениям массива на подотрезки. Петя еще не настолько хорош в программировании, поэтому просит вас помочь.

Поскольку ответ может быть очень большим, вам нужно вывести его остаток по модулю 998244353.

### Формат входных данных

В первой строке находится одно целое число —  $n$  ( $1 \leq n \leq 3 \times 10^5$ ) — размер массива.

Во второй строке находятся  $n$  целых чисел,  $i$ -е из которых равно  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — значения чисел в массиве.

### Формат выходных данных

В единственной строке выведите остаток по модулю 998244353 у суммы привлекательностей по всем разбиениям массива на подотрезки.

### Примеры

стандартный ввод	стандартный вывод
3 2 1 3	2
5 4 4 1 5 3	10

### Замечание

Пояснение к первому примеру:

Массив  $[2, 1, 3]$  можно разбить на подотрезки четырьмя способами:

$[2], [1], [3]$  Красота каждого подотрезка равна 0, значит привлекательность разбиения равна 0.

$[2, 1], [3]$  Красота первого подотрезка  $2 - 1 = 1$ , а второго  $3 - 3 = 0$ , значит привлекательность  $1 \times 0 = 0$ .

$[2], [1, 3]$  Привлекательность  $(2 - 2) \times (3 - 1) = 0$ .

$[2, 1], [3]$  Привлекательность  $3 - 1 = 2$ .

Сумма привлекательностей по всем разбиениям равна 2.

## [A'-A] 4. Наложение цветов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Известно, что если смешать красный и зеленый цвета, то получится желтый цвет.

У вас есть два фильтра, первый фильтр красный, второй зеленый. Каждый фильтр имеет вид прямоугольника размера  $n \times m$ , в каждой клетке которого находится прозрачная пленка. В некоторых клетках пленка бесцветная, в некоторых имеет цвет (красный для первого фильтра, зеленый для второго фильтра).

Известно, что для каждого из фильтров все цветные клетки образуют связную фигуру (от любой цветной клетки можно дойти до любой другой цветной клетки перемещаясь в соседние по стороне цветные клетки).

Если наложить два фильтра друг на друга, то получится прямоугольник размера  $n \times m$ , в клетках которого прозрачные пленки. Каждая клетка может иметь один из следующих цветов:

- бесцветная, при наложении двух бесцветных пленок;
- красная, при наложении красной пленки на бесцветную;
- зеленая, при наложении бесцветной пленки на зеленую;
- желтая, при наложении красной пленки на зеленую.

Вам очень понравился узор из желтых клеток и вы перерисовали его себе в тетрадь.

К сожалению вы потеряли два фильтра. Теперь вы хотите найти какие-нибудь два фильтра, удовлетворяющие условию связности клеток, такие что при наложении получится такое же расположение желтых клеток. Помогите найти такие два фильтра.

### Формат входных данных

В первой строке находится два целых числа  $n, m$  ( $3 \leq n, m \leq 500$ ) — размеры прямоугольника.

В каждой из следующих  $n$  строк находится строка из  $m$  символов. В  $i$ -й строке находится строка  $a_{i1}a_{i2} \dots a_{im}$ . Если  $a_{ij} = '.'$ , то клетка  $(i, j)$  не желтая, если  $a_{ij} = \#$ , то клетка  $(i, j)$  желтая.

**Гарантируется, что в первой строке, первом столбце, последней строке, последнем столбце все клетки не желтые.**

Гарантируется, что существует хотя бы одна желтая клетка.

Гарантируется, что при заданных ограничениях существует хотя бы одно решение.

### Формат выходных данных

Выведите две таблицы из символов размера  $n \times m$ , первая таблица описывает первый фильтр, вторая таблица описывает второй фильтр. Символ '.' обозначает бесцветную клетку, символ '#' обозначает цветную клетку.

В каждой из таблиц клетки '#' должны образовывать связную фигуру. При наложении фильтров должно получаться данное вам расположение желтых клеток.

Если существует несколько возможных решений, выведите любое.

## Пример

стандартный ввод	стандартный вывод
6 6	.....
.....	.####.
.##.#.	.#....
.#....	.#..#.
....#.	.####.
.#.##.	.....
.....	..###.
	.##.#.
	.#..#.
	....#.
	.#.##.
	.###..

## [A'-A] 5. Подготовка ко всеросу

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вы придумали новую гениальную методику подготовки ко всеросу! Подготовка будет занимать  $n$  дней. В  $i$ -й день требуется решить  $p_i$  задач. Поскольку в ходе подготовки вы будете уставать, выполнено что  $p_1 \geq p_2 \geq \dots \geq p_n$ .

Вы не хотите слишком сильно переусердствовать, поэтому вы решили, что начнете подготовку в  $k$ -й день и собираетесь решить в ходе подготовки  $\leq x$  задач суммарно.

Вы будете действовать следующим образом: начиная с  $k$ -го дня по последний, если количество задач  $p_i$ , которое нужно решить в текущий день  $i$ , не превосходит оставшееся количество задач  $x$ , то вы решаете  $p_i$  задач в этот день и  $x$  уменьшается на  $p_i$ . Иначе вы просто пропускаете решение задач в этот день и  $x$  не изменяется.

Перед тем как начать подготовку, вы хотите изменять и тестировать свою методику. Вы хотите обработать  $m$  запросов двух видов:

- 1 к  $x$ : «методика слишком легкая!» Для всех дней  $1 \leq i \leq k$  присвоить  $p_i = \max(p_i, x)$ .
- 2 к  $x$ : «а не бездельник ли я?» Допустим вы начинаете тренироваться в  $k$ -й день и собираетесь решить  $\leq x$  задач суммарно. Какое количество дней вы будете решать задачи?

### Формат входных данных

В первой строке находится два целых числа  $n, m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — количество дней и количество запросов.

Во второй строке находится  $n$  целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq 10^9$ ) — количества задач, которые нужно решать по изначальному плану. Гарантируется, что  $p_1 \geq p_2 \geq \dots \geq p_n$ .

Каждая из следующих  $m$  строк содержит три целых числа  $t, k, x$  ( $1 \leq t \leq 2, 1 \leq k \leq n, 1 \leq x \leq 10^9$ ), описывающие очередной запрос.

Гарантируется, что есть хотя бы один запрос второго типа.

### Формат выходных данных

Для каждого запроса второго типа выведите количество дней, которые вы будете решать задачи.

### Пример

стандартный ввод	стандартный вывод
6 12	2
8 7 5 3 1 1	1
2 3 4	3
2 3 5	1
2 1 12	2
2 6 100	3
1 1 9	4
2 1 12	2
1 5 4	6
2 2 15	
1 3 8	
2 1 24	
2 2 16	
2 1 100	

### Замечание

Изначально  $p = [8, 7, 5, 3, 1, 1]$ .

Рассмотрим первый запрос. Начиная с 3-го дня вы будете решать задачи и планируете решить  $\leq 4$  задач суммарно.

- В 3-й день вы должны решать 5 задач.  $5 > 4$ , поэтому в этот день вы ничего не делаете.
- В 4-й день вы должны решать 3 задачи.  $3 \leq 4$ , поэтому в этот день вы решаете 3 задачи и остается решать  $4 - 3 = 1$  задачу.
- В 5-й день вы должны решать 1 задачу.  $1 \leq 1$ , поэтому в этот день вы решаете 1 задачу и остается решать  $1 - 1 = 0$  задач.
- В 6-й день вы должны решать 1 задачу.  $1 > 0$ , поэтому в этот день вы ничего не делаете.

В итоге вы будете решать задачи 2 дня.

После 5-го запроса  $p = [9, 7, 5, 3, 1, 1]$ .

После 7-го запроса  $p = [9, 7, 5, 4, 4, 1]$ .

После 9-го запроса  $p = [9, 8, 8, 4, 4, 1]$ .

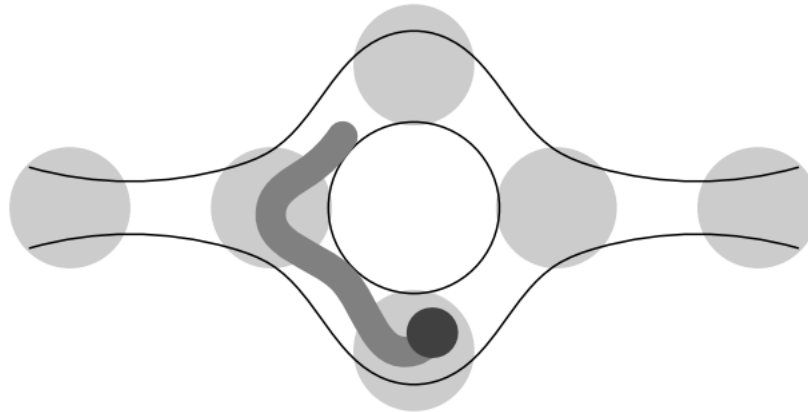
## [A'-A] 6. Шай-Хулуд. Перезапуск франшизы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На планете Арракис обитают огромные песчаные черви, именуемые Шай-Хулуд. Они могут перемещаться по планете, по городам и дорогам, их соединяющим. Дорожная инфраструктура планеты очень специфична. Города связаны дорогами таким образом, что никакие две дороги не соединяют одну и ту же пару городов, никакая дорога не ведёт из города в себя же, а также каждый город лежит не более чем на одном простом цикле.

Червь ползает по дорогам, при этом невозможна ситуация, при которой две разные части тела червя, например, его голова и середина тела, находятся в одной точке. Вам известно начальное положение Шай-Хулуда, который может двигаться только головой вперёд. Также вам дана карта планеты. Ваша задача — вывести кратчайшее расстояние до указанного города.

Длина каждой из дорог равна ровно 1 км. Длина червя равна  $L - \frac{1}{2}$  км. Шай-Хулуд может переползать с дороги на дорогу, находясь головой в каком-то городе, а также его голова не может находиться в том же городе, что и другая часть его тела.



### Формат входных данных

В первой строке ввода дано два целых числа  $n$  и  $k$  ( $1 \leq n, k \leq 100\,000$ ) — количество городов и количество дорог на Арракисе.

В  $i$ -й из следующих  $k$  строк записано по два различных целых числа  $v$  и  $u$  ( $1 \leq u, v \leq n$ ) — номера городов, соединенных  $i$ -й дорогой.

В следующей строке дано целое число  $L$  ( $1 \leq L \leq k$ ) — длина в километрах, на 500 м. большая настоящей длины червя. В следующей строке записано  $L+1$  число: номера городов, между которыми лежат последовательные сегменты Шай-Хулуда, начиная с головы. Между последней парой городов находится хвост Шай-Хулуда, занимающий половину дороги.

Исходное расположение червя корректно

В последней строке записано единственное целое число — город, в который песчаный червь очень хочет переместить свою голову.

### Формат выходных данных

Если Шай-Хулуд может добраться до своей цели, выведите кратчайшую длину пути, то есть количество километров дорог, которое пройдёт голова червя. В противном случае выведите число «-1» (без кавычек).

## Примеры

стандартный ввод	стандартный вывод
6 6 1 2 2 3 3 4 4 5 5 2 4 6 2 5 2 3 1	4
5 5 1 2 2 3 1 3 3 4 4 5 1 4 3 5	1

## [A] 1. Увлекательная прогулка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Серёжа и его друг оказались в хорошо известном им городе. В городе есть  $n$  перекрёстков и  $m$  односторонних улиц, соединяющих эти перекрёстки.

Друзья решили прогуляться. Сейчас они находятся на перекрёстке под номером  $s$ . Сначала друг Серёжи выберет любую улицу и они пойдут по ней к следующему перекрёстку, после чего Серёжа выберет улицу, по которой они пойдут, потом опять друг Серёжи выберет улицу и так далее. Так они будут гулять до тех пор, пока не попадут в вершину, из которой нельзя пройти ни по одной улице. Обратите внимание, что их прогулка может никогда не закончиться.

Но не всё в этом городе так просто... На каждой улице продаётся любимый напиток Серёжи, поэтому каждый раз, когда друзья будут идти по улице, он будет пить этот волшебный напиток. Если они пройдут по  $i$ -й улице от перекрёстка  $v_i$  до перекрёстка  $u_i$ , то Серёжа выпьет  $c_i$  литров этого напитка.

Как было сказано, Серёжа очень любит этот напиток, поэтому хочет выпить его как можно больше. Но ещё больше, Серёжа любит гулять на свежем воздухе, поэтому в первую очередь он хочет, чтобы их прогулка не закончилась, а только во вторую очередь, чтобы он выпил как можно больше волшебного напитка.

Друг Серёжи не хочет долго гулять, поэтому в первую очередь он будет стремиться закончить прогулку. Так же, он заботится о Серёже и не позволит ему выпить слишком много, поэтому вторым приоритетом, он будет выбирать улицы так, чтобы Серёжа выпил как можно меньше этого напитка.

Определите, сколько литров напитка выпьет Серёжа, если оба друга будут действовать оптимально, или сообщите, что их прогулка не закончится.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $m$  и  $s$  ( $1 \leq n \leq 200\,000$ ,  $0 \leq m \leq 200\,000$ ,  $1 \leq s \leq n$ ) — количество перекрёстков, количество улиц и номер перекрёстка, в котором находятся друзья.

Следующие  $m$  строк содержат по три целых числа  $v_i$ ,  $u_i$  и  $c_i$  ( $1 \leq v, u \leq n$ ,  $v_i \neq u_i$ ,  $0 \leq c_i \leq 10^9$ ), что означает, что по  $i$ -й улице можно пройти от перекрёстка  $v_i$  до перекрёстка  $u_i$ , выпив при этом  $c_i$  литров волшебного напитка.

### Формат выходных данных

Выведите «Kaef!» (без кавычек), если прогулка друзей не закончится, и количество литров волшебного напитка, которое они выпьют иначе.



## Примеры

стандартный ввод	стандартный вывод
4 4 1 1 2 0 2 3 0 3 2 0 1 4 1	1
5 5 1 1 2 0 2 3 0 3 4 0 4 3 0 2 5 1	Kaef!
6 6 1 1 2 1 2 3 1000 1 4 10 4 5 1 5 4 1 5 6 10	21

## [A] 2. Разделение королевства

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В королевстве есть  $n$  городов, соединённых между собой  $n - 1$  дорогой, таким образом, что от любого города можно добраться до любого города перемещаясь только по дорогам.

Города королевства будут разделены на два типа. Может быть такое, что не будет ни одного города какого-то из двух типов.

Между каждой парой городов одинакового типа будут построены мосты. Стоимость моста между двумя городами равна длине минимального пути между ними. Длина пути равна количеству дорог в нём.

Вес разделения городов королевства на два типа равен максимальной стоимости построенных мостов или нулю, если не было построено ни одного моста.

Найдите сумму весов всех возможных  $2^n$  разделений городов королевства на два типа по модулю  $10^9 + 7$ .

### Формат входных данных

Первая строка содержит единственное целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — количество городов в королевстве.

Далее следует  $n - 1$  строка, каждая из которых содержит по два целых числа  $v_i$  и  $u_i$  ( $1 \leq v_i, u_i \leq n$ ) — номера городов, которые соединяет  $i$ -я дорога.

### Формат выходных данных

Выведите сумму весов всех возможных разделений городов королевства на два типа по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
4 1 2 2 3 3 4	38
6 1 2 2 3 1 4 1 5 5 6	224

## [A] 3. Теодор Рузвельт - 2

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7 секунд
Ограничение по памяти:	1024 мегабайта

«Теодор Рузвельт» - флагман военно-морского флота Кукуляндии, уже переживший атаку флатляндцев. Из-за пробоин, полученных от баллистических ракет, корабль теперь может потонуть даже от не очень сильных волн! Метеорологи, работающие на корабле, засекали все волны, направляющиеся в сторону корабля. Каждая волна является прямой, заданной уравнением  $ax + by + c = 0$ . Сами по себе волны не страшны «Теодору Рузвельту», но если в одной точке встречаются две волны сразу, то их суммарная сила уже может потопить корабль. Помогите капитану корабля среди всех  $\frac{n(n-1)}{2}$  опасных точек найти расстояние до  $k$ -й по удаленности от корабля. Сам корабль находится в точке  $(0, 0)$ . Гарантируется, что никакие две прямые, задающие волны, не являются параллельными.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  ( $2 \leq n \leq 5 \cdot 10^4, 1 \leq k \leq \frac{n(n-1)}{2}$ ) – количество волн и параметр  $k$ .

В следующих  $n$  строках даны по три целых числа  $a_i, b_i$  и  $c_i$  ( $-1000 \leq a_i, b_i, c_i \leq 1000, a_i^2 + b_i^2 \neq 0$ ), описывающие  $i$ -ю волну.

### Формат выходных данных

В единственной строке выведите ответ на задачу. Ваш ответ будет считаться правильным, если его относительная или абсолютная погрешность от правильного не превосходит  $10^{-4}$ .

### Примеры

стандартный ввод	стандартный вывод
3 2 3 -1 1 3 2 1 0 1 -2	2.02758751009940629828
5 7 1 -7 -4 7 3 7 8 9 8 6 2 -6 4 4 0	2.47487373415291500578

## [A] 4. Супершанс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Супер-Кота день рождения, и Леди Баг не знает что ему подарить! Чтобы найти хороший подарок, она использовала супершанс, и получила перестановку  $p$  длины  $n$ .

Однако вся перестановка не поместится в подарочную коробку, поэтому перестановку надо разбить на  $k$  непустых и непересекающихся подотрезков, чтобы положить один подотрезок в одну коробку. Конечно, каждый элемент перестановки должен попасть ровно в один из отрезков.

Супер-Кот будет открывать коробки в каком-то порядке, и по очереди доставать отрезки. Он поставит эти отрезки по порядку друг за другом, и таким образом, соберет перестановку  $q$ .

Леди Баг знает, что Супер-Кот очень любит лексикографически минимальные перестановки.

Однако, Бразжник терпеть не может Леди Баг и Супер-Кота, поэтому он перемешает коробки так, чтобы полученная Супер-Котом перестановка была лексикографически максимальной среди всех перестановок, которые он может получить. То есть Бразжник выберет последовательность, в которой Супер-Кот будет открывать коробки.

Помогите Леди Баг разбить перестановку на отрезки так, чтобы итоговая перестановка  $q$ , которую получит Супер-Кот, была лексикографически минимальной.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 200\,000$ ) — количество элементов в перестановке и количество отрезков, на которое надо разбить перестановку.

Вторая строка содержит  $n$  целых чисел  $p_1, p_2, p_3, \dots, p_n$  ( $1 \leq p_i \leq n$ ) — элементы перестановки  $p$ . Гарантируется, что  $p$  является перестановкой.

### Формат выходных данных

Выведите  $n$  целых чисел  $q_1, q_2, q_3, \dots, q_n$  — перестановку, которую получит Суперкот, если Ледибаг будет действовать оптимально.

### Примеры

стандартный ввод	стандартный вывод
4 2 1 2 3 4	2 3 4 1
5 3 1 4 3 5 2	3 5 2 1 4
7 4 1 6 4 5 2 3 7	4 5 3 7 2 1 6

## [A] 5. За деньги да

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Инстасамка и ее парень Монеукеп учатся петь. Для этого они поют ноты по очереди, при этом каждая нота обозначается цифрой от 1 до 9. Сегодня они тренируют восходящие мелодии, поэтому мелодия никогда не должна спускаться вниз. Таким образом, каждую мелодию можно представить как число в десятичной записи, в котором цифры не убывают слева направо. Назовем такое число десятичным представлением мелодии.

Например, числа 123 и 1155899 являются корректными десятичными представлениями мелодий, а числа 54, 20 и 8891 — нет.

Инстасамка и Монеукеп хотят спеть несколько мелодий, так чтобы сумма их десятичных представлений была равна  $n$ .

Однако, они не очень любят учиться, поэтому хотят спеть как можно меньше мелодий. Помогите им узнать, какое минимальное количество мелодий нужно спеть, что бы сумма их десятичных представлений была равна  $n$ .

### Формат входных данных

Вводится одно число  $n$  ( $1 \leq n \leq 10^{500\,000}$ ) — итоговая сумма, которую надо получить.

### Формат выходных данных

Выведите одно число  $k$  — минимальное количество мелодий, которые нужно спеть Инстасамке и Монеукеп, чтобы сумма их десятичных представлений давала  $n$ .

### Примеры

стандартный ввод	стандартный вывод
51	2
13579	1
62346161	4

### Замечание

В первом примере подойдут следующие мелодии 48 и 3, их сумма равна 51.

Во втором примере само число 13579 является десятичным представлением мелодии, поэтому ответ равен 1.

## [A] 6. Шаи-Хулуд. Финальная битва

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На планете Арракис обитают огромные песчаные черви, именуемые Шаи-Хулуд. Они могут перемещаться по планете, по городам и дорогам, их соединяющим. Дорожная инфраструктура планеты очень специфична. Города связаны дорогами таким образом, что никакие две дороги не соединяют одну и ту же пару городов, никакая дорога не ведёт из города в себя же, а также каждый город лежит не более чем на одном простом цикле.

Недавно прошедшая песчаная буря уничтожила всю дорожную инфраструктуру планеты Арракис, поэтому черви вернулись к традиционному для них способу перемещения под землей. Подземные просторы планеты можно представить как трехмерное пространство с осями  $A$ ,  $B$  и  $C$  перпендикулярными друг другу, поверхность планеты находится на плоскости с координатами 0 по оси  $C$ .

Как известно, черви состоят из нескольких сегментов. Каждый сегмент является усеченным конусом, который вплотную прилегает к предыдущему и следующему сегменту. Более формально, пусть Шаи-Хулуд состоит из  $n$  сегментов. Тогда Шаи-Хулуда можно представить следующим образом: Представим себе ломанную, построенную в пространстве, имеющую  $n + 1$  вершину и соответственно  $n$  сегментов, каждый из которых имеет длину  $L$  км. Угол между любыми двумя сегментами ломанной не менее  $90$  градусов. В каждой вершине Шаи Хулуда строится окружность, радиус окружности в  $i$ -й вершине равен  $r_i$ . Гарантируется, что  $d_i < L$ . Известно, что угол между плоскостью окружности и выходящими из неё сегментами всегда больше  $45$  и меньше  $90$  градусов.

По каждым двум соседним окружностям (их центры соединены сегментом ломанной) строится усеченный конус, который проходит через эти две окружности. Внутренностью  $i$ -го сегмента червя называется всё внутреннее пространство усеченного конуса, построенного между  $i$ -й и  $i + 1$ -й окружностями. Известно, что для любых двух сегментов Шаи-Хулуда, их внутренности не пересекаются.

За один шаг червь может продвинуться вперед на один сегмент. В процессе движения его последний сегмент удаляется, перед первым сегментом добавляется новый. Таким образом новая ломанная имеет  $n + 1$  вершину. Далее, в вершине  $i$  строится окружность с радиусом  $r_i$ , направление плоскости окружности остается таким же, каким оно было в этой точке до движения, а направление окружности в первой точке выбирается червем. После движения должны выполняться все правила построения червя, в том числе правило, что никакие два сегмента Шаи-Хулуда не пересекаются. Обратите внимание, что сначала удаляется последний сегмент, а только потом добавляется новый, то есть удаленный и добавленные сегменты могут пересекаться.

Известно начальное расположение червя и то, в которое он хочет прийти. Шаи-Хулуд хочет это сделать за минимальное количество действий, и поэтому он обратился к вам за помощью. К сожалению, в то время, когда до вас дошло это сообщение, вы были заняты отлаживанием кода задачи для движения червя по старой системе дорог, поэтому вы его проигнорировали, и правильно сделали!

Сам Шаи-Хулуд не справился с этой задачей, а поэтому остался стоять на месте. Чтобы ему не было скучно, к нему подполз другой червь и они решили сыграть в Арракиские шахматы. Эти шахматы устроены особым образом: доска в этих шахматах бесконечно большая и фигур очень много. Для удобства, занумеруем все клетки доски координатами по оси  $x$  и  $y$ .

Исходно первый игрок имеет по одной фигуре в каждой клетке  $(x, y)$ , такой что  $-X_1 \leq x \leq -1$  и  $-Y_1 \leq y \leq -1$ . Второй игрок имеет по одной фигуре в каждой клетке  $(x, y)$ , такой что  $1 \leq x \leq X_1$  и  $1 \leq y \leq Y_1$ . В клетке с координатами  $(x, y)$  находится фигура одного из 6 типов:

- К — Король. Король может ходить с клетки  $(x, y)$  на любую клетку  $(x_1, y_1)$ , такую что  $|x - x_1| \leq 1$  и  $|y - y_1| \leq 1$ , при этом во время хода данной фигурой, она не может оставаться на месте. Гарантируется, что у каждого игрока ровно одна фигура этого типа.

- **R** — Ладья. Ладья может ходить с клетки  $(x, y)$  на любую клетку  $(x_1, y_1)$ , такую что  $x = x_1$  или  $y = y_1$ , при этом во время хода данной фигурой, она не может оставаться на месте.
- **B** — Слон. Слон может ходить с клетки  $(x, y)$  на любую клетку  $(x_1, y_1)$ , такую что  $|x - x_1| = |y - y_1|$ , при этом во время хода данной фигурой, она не может оставаться на месте.
- **Q** — Ферзь. Королева может ходить по правилам и слона, и ладьи.
- **k** — Конь. Конь может ходить с клетки  $(x, y)$  на любую клетку  $(x_1, y_1)$ , такую что  $|x - x_1| + |y - y_1| = 3$  и  $x_1 \neq x$  и  $y_2 \neq y$ , при этом во время хода данной фигурой, она не может оставаться на месте.
- **P** — Пешка. Пешка может ходить с клетки  $(x, y)$  на любую свободную клетку  $(x_1, y_1)$ , такую что  $|x| - 1 \leq |x_1| \leq |x|$  и  $|y| - 1 \leq |y_1| \leq |y|$  и  $x = x_1$  или  $y = y_1$ . Так же пешка может ходить на любую занятую клетку, такую что  $|x| - 1 \leq |x_1| \leq |x| + 1$  и  $|y| - 1 \leq |y_1| \leq |y| + 1$  и  $x \neq x_1$  и  $y \neq y_1$ . При этом во время хода данной фигурой, она не может оставаться на месте. При достижении пешкой клетки, у которой хотя бы одна координата равна нулю, она может стать любой фигурой из списка: Ладья, Слон, Конь, Ферзь.

Игроки играют по очереди, за ход игрок может подвинуть любую свою фигуру. Если игрок двигает свою фигуру на чужую фигуру, то эта чужая фигура считается съеденной и покидает игру. Как в обычных шахматах, игроки во время движения фигур не могут их перемещать над другими фигурами. Игрок считается победившим, если съел Короля другого игрока. Ничьей считается патовая ситуация, при которой сторона, имеющая право хода, не может им воспользоваться, так как все её фигуры и пешки, находящиеся на доске, лишены возможности сделать ход по правилам, причём король не находится под шахом (то есть игрок не может сделать ход, после которого король этого игрока не будет атакованным).

Шаи-Хулуды начали играть в шахматы, и уже сделали несколько ходов, после чего получили следующую ситуацию: У игрока 1 осталось всего 5 слонов,  $i$ -й из них расположен в клетке с координатами  $(x_i, y_i)$ . У второго игрока остался лишь один король, расположенный в клетке с координатами  $(a, b)$ . Теперь первому игроку требуется разработать стратегию, которая приведет к его победе или пату. Отсутствие короля у первого игрока можно игнорировать. Сейчас ход первого игрока.

Мы надеемся, что в этот раз вы не проигнорируете очередную просьбу Шаи-Хулуда, иначе он на вас обидится и в следующем году пошлет 12 разных просьб!

## Протокол взаимодействия

**Это интерактивная задача.**

Первая строка ввода будет содержать единственное целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных.

Каждый набор входных данных начинается с пяти пар целых чисел  $x_i, y_i$ , и пары чисел  $a, b$  записанных на одной строке, где первые пять пар задают координаты слонов первого игрока, а последняя пара задаёт координаты короля второго игрока. Все эти числа не превосходят по модулю  $10^6$ .

Далее начинается игра. Ваш ход состоит из четырёх целых чисел  $x_1 y_1 x_2 y_2$  на одной строке — это начальные и конечные координаты слона, которым первый червь должен сделать ход.

В ответ вы получаете четыре целых числа  $x_1 y_1 x_2 y_2$  на одной строке — это начальные и конечные координаты короля, которым второй червь сделал ход.

Когда поставлен мат/пат, интерактор выдаёт четыре нуля, после чего переходит к следующему тестовому случаю (если он есть). Если ваша программа решает, что не существует способа поставить мат или пат, она должна вывести четыре нуля вместо первого хода, в этом случае интерактор немедленно переходит к следующему тестовому случаю (если он есть). Координаты перемещаемых фигур не должны превосходить  $10^9$  по модулю.

Подробности можно увидеть в примере.

Если мат/пат поставить возможно, ваша программа должна сделать это не более чем за 50 ходов, иначе ответ будет считаться неверным. Перед вами не стоит задачи найти оптимальное решение, но гарантируется, что в любой позиции, в которой решение существует, можно достигнуть цели не более чем за 50 ходов.

Если мат/пат поставить невозможно, ваша программа должна немедленно ответить строчкой из четырёх нулей (запрещается делать ходы), иначе ответ будет сочтён неверным.

Всегда гарантируется, что входная позиция корректна, то есть никакие две фигуры не находятся в одной клетке и королю не поставлен шах.

После каждой выведенной строки следует очищать буфер вывода.

**Так как задача является интерактивной, то если ваше решение неправильное, вы можете получить по нему любой вердикт не являющийся ОК.** То есть вместо любого вердикта (WA, RE, TL и другие) вы можете получить любой другой, не соответствующий правильному решению. Если ваше решение правильное, вы точно получите вердикт ОК.

## Пример

стандартный ввод	стандартный вывод
4	
2 2 3 3 4 4 5 5 6 6 8 6	0 0 0 0
2 2 2 3 2 5 2 6 3 3 4 6	3 3 2 4
0 0 0 0	
2 3 3 4 5 6 6 7 7 8 6 4	0 0 0 0
4 3 5 3 4 7 5 7 7 5 4 5	7 5 6 6
0 0 0 0	

## Замечание

Пустые строки добавлены лишь для того, чтобы показать последовательность событий.

В первом и третьем тестах все пять слонов одного цвета. Во втором тесте можно поставить за один ход мат, а в четвёртом — пат.