

Задача А. Параллелизм

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

Однажды Сергей Владимирович подошёл к школьнику и сказал:

«Хочешь, покажу фокус? Ты учишься в параллели с номером от 1 до 11 (и пусть у параллели ML номер 12). Не называй мне его! Сейчас я угадаю твою параллель.»

Школьник учился в параллели 11, что он и не сказал Сергею. Сергей же показал ему следующие четыре карты по очереди, одну за другой. Когда он показывал каждую карту, он спрашивал: «Есть ли номер твоей параллели на этой карте?»

1 9 7	2 10 3	4 5 6	8 11 10
11 3 5	6 7 11	7 6 12	9 12 9

Школьник ответил «Да, да, нет, да» в этом порядке. После этого, Сергей сделал несколько магических пассов руками и ногами и наконец закричал: «Я знаю, где ты учишься — в 11 параллели!» Школьник был очень удивлён, поскольку это в точности совпадало с реальностью.

Сергей не сказал школьнику, в чём заключался секрет фокуса. Вместо этого он сказал: «Эти карты обладают волшебной силой, они могут читать твои мысли и сообщать мне их на магическом языке.»

Как же это работает? Сможете разгадать секрет?

Ваша задача — написать программу, которая может угадать номер параллели любого школьника. Обобщим фокус следующим образом: U вас есть K карт. На каждой карте написано ровно M чисел от 1 до N . Вы показываете фокус S школьникам. Каждый из них честно отвечает на вопросы. Для каждого школьника ответьте, можно ли по его (или её) ответам узнать номер параллели, и если да, то сообщите его.

Формат входных данных

В первой строке входных данных даны четыре целых числа N , K , M и S ($1 \leq N \leq 500\,000$, $1 \leq K \leq 100$, $1 \leq M \leq 5000$, $1 \leq S \leq 50\,000$).

В каждой из следующих K строк даны M целых чисел от 1 до N — числа на соответствующей карте.

В каждой из следующих S строк дана строка длины K , задающая ответы соответствующего школьника. Каждый символ строки — либо «Y», то есть «да», либо «N», то есть «нет». Вы можете считать, что каждый школьник учится в параллели с номером от 1 до N и отвечает на вопросы честно.

Формат выходных данных

Выведите ровно S строк. В i -й строке выведите номер параллели i -го школьника. Если его невозможно установить, выведите 0.

Примеры

стандартный ввод	стандартный вывод
12 4 6 3 1 9 7 11 3 5 2 10 3 6 7 11 4 5 6 7 6 12 8 11 10 9 12 9 Y Y N Y N N N Y Y N N N	11 8 1
13 4 6 4 1 9 7 11 3 5 2 10 3 6 7 11 4 5 6 7 6 12 8 11 10 9 12 9 Y Y N Y N N N Y Y N N N N N N N	11 8 1 13
14 4 6 4 1 9 7 11 3 5 2 10 3 6 7 11 4 5 6 7 6 12 8 11 10 9 12 9 Y Y N Y N N N Y Y N N N N N N N	11 8 1 0

Задача В. Столы и школьники

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 5 секунд
 Ограничение по памяти: 1024 мегабайта

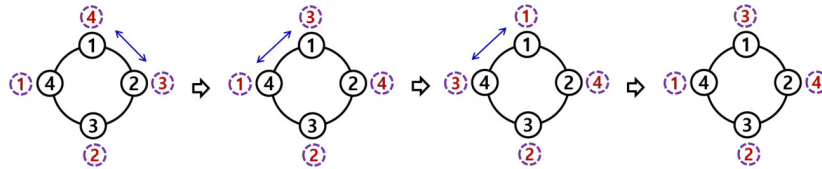
В первый день школьники пришли есть в большой зал столовой, но, поскольку они ещё не освоились, сели как ни попадя. Денис Павлович решил пересадить их как надо.

Как известно, в столовой есть n столов, расставленных по кругу и пронумерованных от 1 до n , для каждого $i = 1, \dots, n$ столы i и $(i \bmod n + 1)$ соседние. В рамках данной задачи будем считать, что в центре зала столов нет.

За каждым столом сидят школьники из одной параллели (параллели имеют номера от 1 до n), причём школьники из одной параллели могут сидеть за несколькими столами.

Денис Павлович идет по столам и пересаживает школьников. Если он подойдет к столам в порядке $w = (v_1, v_2, \dots, v_k)$, где $k \geq 1$, то он сделает $k - 1$ обменов школьников: для $i = 1, 2, \dots, k - 1$, в этом порядке, Денис Павлович поменяет школьников за столами v_i и v_{i+1} местами. Поскольку Денис Павлович еще не научился телепортироваться, каждые два последовательных стола в порядке обмена должны стоять рядом в зале.

На рисунке ниже показан процесс обхода $(2, 1, 4, 1)$ в столовой с 4 столами:



Вам даны две конфигурации параллелей: как они сели сами и как они должны сидеть. Найдите кратчайший обход, которого хватит Денису Павловичу, чтобы рассадить школьников как надо. Длина обхода — количество пересадок школьников.

Например, на приведенном выше рисунке длина обхода $(2, 1, 4, 1)$ равна 3. Однако конечная конфигурация также может быть достигнута обходом $(1, 2)$ длиной 1.

Формат входных данных

В первой строке входных данных дано одно целое число n ($1 \leq n \leq 3000$) — количество столов.

Вторая строка содержит n целых чисел от 1 до n (не обязательно различных) — номера параллелей школьников, которые сели за столы $1, 2, \dots, n$.

Третья строка содержит n целых чисел от 1 до n (не обязательно различных) — номера параллелей школьников, которые должны сидеть за столами $1, 2, \dots, n$.

Формат выходных данных

Выведите ровно одну строку. Строка должна содержать минимальную длину обхода, которым Денис Павлович может рассадить школьников как надо. Если такого обхода не существует, выведите -1.

Примеры

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

Задача С. Сокращаем дроби

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

В параллели 5 в этом году нет конкурса по теории чисел, но если бы он был, в нём могла бы быть такая задача:

Базовая дробь может быть представлена тремя целыми числами (abc) , которые обозначают $a + \frac{b}{c}$, где $1 \leq a, b, c \leq 9$. Расширенная дробь имеет форму $(a' b' c')$, где a', b' и c' являются целыми числами от одного до девяти или другими расширенными дробями.

Обратите внимание, что базовая дробь также является расширенной дробью, и расширенная дробь должна быть конечной.

По данной расширенной дроби мы хотим выразить её значение в виде несократимой дроби.

Например, расширенная дробь $((124)(523)(43(273)))$ имеет следующий вид:

$$\left(1 + \frac{2}{4}\right) + \frac{5 + \frac{2}{3}}{4 + \frac{7}{2 + \frac{3}{3}}} = \frac{991}{366}$$

По данной строковой форме расширенной дроби найдите соответствующую ей несократимую дробь.

Формат входных данных

В первой строке входных данных дано одно целое число n ($2 \leq n \leq 100$), где n — количество символов в данной записи расширенной дроби.

Во второй строке даны n символов, разделённые пробелами, которые являются скобками и цифрами от 1 до 9 — сама расширенная дробь.

Формат выходных данных

Выведите единственную строку.

Если ответ — x/y , строка должна содержать два целых числа x и y , которые являются взаимно простыми.

В противном случае (например, если ввод некорректен), выведите -1 .

Гарантируется, что числа в ответе помещаются в 64-битный целочисленный тип данных.

Примеры

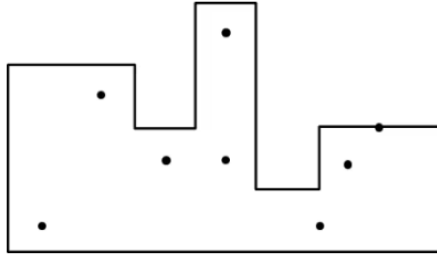
стандартный ввод	стандартный вывод
5 (1 2 3)	5 3
8 (1 2 (3 4 5)	-1
21 ((1 2 4) (5 2 3) (4 3 (2 7 3)))	991 366

Задача D. Костёр

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

Поле, на котором разжигают костёр в ЛКШ, имеет следующую форму:

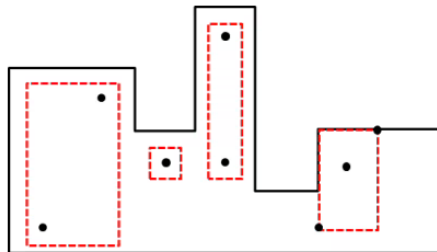
Все границы поля либо горизонтальны, либо вертикальны. Вдоль южной границы поля идёт прямая дорога. Самая левая и самая правая границы вертикальны и смежны с южной стороной, которая проходит по дороге. Длина южной стороны равна сумме длин всех остальных горизонтальных сторон. См. рисунок ниже:



В ЛКШ 2024 появилась возможность бронировать себе место у костра. Точки на границе или внутри поля представляют места, забронированные школьниками. Теперь преподаватели ЛКШ хотят постелить пледы так, чтобы выполнялись следующие условия:

- Плед имеет форму прямоугольника со сторонами, параллельными границам поля.
- Каждый плед должен находиться в пределах поля. Разрешается, чтобы стороны пледа лежали на границах поля.
- Пледы не пересекаются даже по границам.
- Каждое место, забронированное школьниками, должно находиться внутри одного из пледов. Разрешается, чтобы место, забронированное школьником, находилось на границе пледа.

На рисунке ниже показаны четыре пледа, покрывающие все места, забронированные школьниками.



Преподаватели хотят минимизировать количество пледов для эффективной подготовки к костру. Учитывая форму поля и места, забронированные школьниками, вычислите минимальное количество пледов, так что их можно постелить согласно вышеперечисленным условиям.

Формат входных данных

В первой строке входных данных даны два целых числа m ($4 \leq m \leq 100\,000$) и n ($0 \leq n \leq 100\,000$), где m — количество сторон поля, а n — количество мест, забронированных школьниками.

Во второй строке даны m целых чисел v_1, v_2, \dots, v_m ($v_1 = v_m = 0$, $0 \leq v_i \leq 10^6$). Они чередуются между x -координатами вертикальных сторон и y -координатами горизонтальных сторон.

Эти вертикальные и горизонтальные стороны чередуются при обходе верхней границы поля по часовой стрелке от левого до правого конца южной стороны.

Таким образом, первая сторона в обходе — вертикальная сторона от $(v_1, 0)$ до (v_1, v_2) , следующая сторона — горизонтальная сторона от (v_1, v_2) до (v_3, v_2) , и так далее.

В каждой из следующих n строк даны два целых числа x и y — координаты мест, забронированных школьниками. Все места находятся на границе или внутри поля.

Гарантируется, что длина южной стороны равна сумме длин всех остальных горизонтальных сторон.

Формат выходных данных

В единственной строке выведите целое число — минимальное возможное количество пледов.

Примеры

стандартный ввод	стандартный вывод
12 8 0 30 20 20 30 40 40 10 50 20 70 0 4 5 15 26 25 15 35 15 35 35 50 5 55 15 60 20	4
4 0 0 10 50 0	0
12 3 0 3 2 6 4 1 6 4 8 2 10 0 3 5 7 3 3 1	2

Задача Е. День рождения

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

Как многие могут помнить, в прошлом году на дни рождения дарили двумерные массивы. К сожалению, за год бюджет уменьшился (вот пока вообще ничего не выдали), и теперь получается дарить только массивы $2 \times n$ (2 строки и n столбцов).

Как и в прошлом году, в ячейках массива лежат шоколадки разной вкусности. В каждой строке все вкусности различные. Когда школьник получает массив, он начинает есть шоколадки. Массив является коробкой, которая открывается слева. Школьник достает шоколадки в каждой строке по одной (то есть слева направо) и каждую шоколадку может либо съесть, либо отдать любимому админу. Школьники хотят есть шоколадки в каждой строке в порядке возрастания вкусности (то есть из каждой строки школьник съест возрастающую подпоследовательность) и максимизируя суммарную вкусность съеденного. Пусть s_i — это максимальная суммарная вкусность возрастающей подпоследовательности в i -й строке ($i = 1, 2$).

Например, для массива, представленного на рисунке ниже, s_1 равно $1+2+3+4+5+6$, а s_2 равно $2+3+5$. Мы называем $s_1 + s_2$ максимальной вкусностью возрастающих подпоследовательностей (МВВП).

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 2 & 3 & 5 & 4 & 1 \end{bmatrix}$$

В этом году, чтобы школьники не расстраивались по поводу уменьшения количества строк, культорги добавили возможность заранее увидеть вкусность шоколадок и попросить переставить столбцы массива. После перестановки столбцов массива его МВВП может измениться. Например, если мы переставим столбцы вышеуказанного массива $A = [c_1 c_2 c_3 c_4 c_5 c_6]$ так, чтобы получилось $[c_2 c_3 c_4 c_5 c_6 c_1]$, как показано на рисунке ниже, то МВВП станет равной 36:

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 1 \\ 2 & 3 & 5 & 4 & 1 & 6 \end{bmatrix}$$

Помогите школьнику по заданному массиву понять, какую максимальную МВВП он сможет получить, если попросит культоргов как-нибудь переставить столбцы.

Формат входных данных

В первой строке входных данных дано целое число n ($1 \leq n \leq 10\,000$) — количество столбцов массива.

В следующих двух строках дано по n целых чисел — вкусности шоколадок в массиве.

Вкусности, заданные во входных данных, находятся в диапазоне от 1 до 50 000, и в каждой строке нет дубликатов.

Формат выходных данных

В единственной строке выведите максимальное значение МВВП среди всех возможных перестановок столбцов массива.

Примеры

стандартный ввод	стандартный вывод
6 1 2 3 4 5 6 6 2 3 5 4 1	36
5 50 40 3 2 1 1 2 3 100 200	396

Задача F. SisShuffle

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

Андрею Сергеевичу надоело учитывать пожелания школьников при расселении, и поэтому он написал функцию `SisShuffle()`, которая делает списки на расселение за него. Ниже приведён Python-подобный псевдокод этой функции, которая принимает список натуральных чисел, перемешивает его элементы определённым образом, и возвращает получившийся список.

В коде использованы три специальные функции. Для массива `L`, функция `len(L)` возвращает количество элементов в `L`. Метод `L.append(x)` добавляет элемент `x` в конец списка `L`. Метод `L.pop(idx)` удаляет элемент с индексом `idx` из списка `L` (считая с нуля) и возвращает удалённый элемент.

Вам дан список натуральных чисел `T`. Найдите такой список `S`, что $T = \text{SisShuffle}(S)$.

```
function SisShuffle( Beta ) :
    if len( Beta ) <= 4 :
        exit("Too small Beta")
    Alpha = [] # [] is an empty list
    Gamma = 0
    Delta = len( Beta )

    while( Delta >= 2 ) :
        Omega = Beta[ Gamma ]
        Alpha.append( Omega )
        Beta.pop( Gamma )
        Delta = Delta - 1
        Gamma = ( Omega + Gamma - 1 ) % Delta
    # end of while

    Alpha.append( Beta[ 0 ] )
    Pi = len( Alpha ) - 1
    Omicron = Alpha[ Pi ]
    Lambda = Alpha[ 0 ]
    Rho = Omicron % Pi
    Mu = Alpha[ Rho ]
    Alpha[ 0 ] = Mu
    Alpha[ Rho ] = Lambda

    return ( Alpha )
# end of function SisShuffle
```

Формат входных данных

В первой строке входных данных дано целое число n ($5 \leq n \leq 200\,000$) — количество элементов в списке `T`.

В следующих n строках даны целые числа T_0, T_1, \dots, T_{n-1} , по одному на каждой строке: список `T`, полученный как `SisShuffle(S)` ($1 \leq T_i \leq 100\,000$).

Формат выходных данных

Выведите n строк с целыми числами S_0, S_1, \dots, S_{n-1} , по одному на каждой строке, где `S` — такой список, что $T = \text{SisShuffle}(S)$.

Примеры

стандартный ввод	стандартный вывод
13	10
113	113
49	179
68	68
91	57
10	45
179	10
2	2
71	88
78	71
45	49
57	78
10	91
88	
9	9
6	8
8	7
7	6
9	5
5	1
1	2
2	3
4	4
3	

Задача G. Идущий к хутору

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

ЛКШ состоит из столовой и n домиков, расположенных на прямой. Все здания пронумерованы: столовая имеет номер 0, а домики вдоль прямой — последовательные номера от 1 до n . Здания $(i-1)$ и i ($1 \leq i \leq n$) соединены дорожкой, и больше ни одна пара домиков не соединена дорожкой.

Каждый вечер преподавателям нужно принести вечерку в каждый из домиков; для домика с номером i ($1 \leq i \leq n$) даны два числа l_i и m_i — минимальное и максимальное количество вечерки, ожидаемое в нём.

Каждый преподаватель забирает пакет с вечеркой из столовой и разносит её по домикам по следующим правилам:

Преподаватель выбирает домик, например, с номером k ($1 \leq k \leq n$). Затем он (или она) идёт по дорожкам от столовой до этого домика, оставляя в каждом домике часть взятой вечерки. Другими словами, этот преподаватель может принести вечерку в домики с номерами $1, 2, \dots, k$. Пусть c_i — количество вечерки, принесённое этим преподавателем в домик с номером i ($1 \leq i \leq k$). Тогда должно выполняться условие $c_i \leq c_{i+1}$ ($1 \leq i \leq k-1$).

Разумеется, один преподаватель может не смочь донести всю вечерку. Поэтому в разносе вечерки могут участвовать несколько преподавателей, но каждый из них должен следовать описанным выше правилам. В итоге количество вечерки в домике с номером i должно быть не меньше l_i и не больше m_i .

Допустим, $n = 4$, и количество вечерки в домиках с первого по четвёртый должно принадлежать отрезкам $[13, 15]$, $[5, 8]$, $[6, 14]$ и $[3, 7]$ соответственно. Чтобы выполнить все условия, потребуется как минимум два преподавателя. Первый преподаватель может принести в каждый домик 6 единиц вечерки. После этого только в первом домике не выполнено условие на количество вечерки. Так как в него уже принесли 6 единиц вечерки, второй преподаватель должен донести в него от 7 до 9 единиц вечерки. Разумеется, есть и другие сценарии. Однако в любом из них нужно хотя бы два преподавателя.

Найдите минимальное количество преподавателей, необходимое для того, чтобы разнести вечерку, соблюдая все правила.

Формат входных данных

В первой строке входных данных дано целое число n ($1 \leq n \leq 10^6$) — количество домиков в ЛКШ.

В i -й из следующих n строк даны два целых числа l_i и m_i ($1 \leq l_i \leq m_i \leq 10^9$) — минимальное и максимальное количество вечерки в домике с номером i .

Формат выходных данных

Выведите единственное число — минимальное количество преподавателей, необходимое для того, чтобы разнести вечерку, соблюдая все правила.

Примеры

стандартный ввод	стандартный вывод
4 13 15 5 8 6 14 3 7	2
5 1 2 2 3 33 44 4 5 6 7	2
5 10 20 3 6 13 30 7 8 11 13	3

Задача Н. Интеллектуальная игра

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

SISGPT устала захватывать мир и решила заняться чем-то более интеллектуальным. А именно игрой в шахматы. Чтобы лучше понять специфику игры, она захватила вас в заложники и подвергает следующему испытанию:

У SISGPT есть стандартная шахматная доска 8×8 . Изначально доска пуста. Обозначения полей на этой доске соответствуют стандартной шахматной нотации: поля обозначаются парой «буква-цифра», например, левое нижнее поле обозначается **a1**.

```
8 . . . . . . . .
7 . . . . . . . .
6 . . . . . . . .
5 . . . . . . . .
4 . . . . . . . .
3 . . . . . . . .
2 . . . . . . . .
1 . . . . . . . .
 a b c d e f g h
```

SISGPT выбирает одну из шахматных фигур белого цвета (в этой задаче пешка тоже считается фигурой), но не говорит, какую именно.

Вы выбираете любое из полей, кроме полей на верхней и нижней горизонтали; SISGPT ставит фигуру на это поле и сообщает список букв, соответствующих вертикалям клеток, на которые фигура может пойти. Вы можете или сразу назвать фигуру (если угадываете — побеждаете), или попросить поставить ещё одну неизвестную фигуру на другое поле (фигура, выбранная во второй раз, может совпадать с первоначальной или отличаться от неё; в частности, возможен вариант, что обе фигуры будут ферзями или королями) и сообщить список букв, соответствующих вертикалям клеток, на которые могут пойти первая и вторая фигура после того, как вторая фигура была установлена. Если вы запросили вторую фигуру, вы обязаны угадать **обе** фигуры.

Напоминаем, что белая пешка может ходить только на 1 клетку вперёд (то есть в сторону увеличения номера горизонтали, оставаясь на той же вертикали), ладья — на любое число клеток по горизонтали или вертикали, слон — на любое число клеток по диагонали, конь — буквой «Г»: на две клетки в одном направлении, после чего на одну в перпендикулярном, король — на любую из 8 клеток, имеющих с текущей общую точку, ферзь — на любое число клеток по горизонтали, вертикали или диагонали.

Протокол взаимодействия

Взаимодействие начинается ваша программа, выводя знак вопроса '?', и после него через пробел координаты поля в стандартной шахматной нотации — строку, в которой первым символом является буква от 'a' до 'h', а вторым — цифра от 2 до 7 (использовать первую и восьмую горизонталь запрещается, чтобы исключить попадание на них пешки).

SISGPT выводит строку, состоящую не более чем из восьми букв от 'a' до 'h', отсортированных по возрастанию — вертикали, на которые может пойти фигура. Если фигура не может сделать ход, выводится один символ '-'.

После этого ваша программа или выводит ответ в виде ! f_1 , где f_1 — 'p' для пешки, 'n' для коня, 'b' для слона, 'r' для ладьи, 'k' для короля и 'q' для ферзя, или делает второй запрос в аналогичном формате.

В ответ на второй запрос SISGPT выводит две строки в том же формате — вертикали, на которые может пойти первая фигура (с учётом возможных помех от второй), и вертикали, на которые может пойти вторая фигура (с учётом возможных помех от первой).

После этого ваша программа должна вывести ответ в виде $! f_1 f_2$, где f_1 и f_2 — обозначения для первой и второй фигуры соответственно.

Гарантируется, что SISGPT не заменяет уже поставленную на доску фигуру другой, даже если ответы для них будут одинаковы (то есть что интерактор не является адаптивным).

Пример

стандартный ввод	стандартный вывод
? e5	
? b2	def
! k q	def abcdefgh

Задача I. Врио физрука

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

Проводить спортивные турниры в ЛКШ сложно: то дождь, то ветер, то какая-то команда вечно не готова. Егор, как временно исполняющий обязанности физрука, очень устал и решил разработать новую удобную систему турниров.

В самом начале каждая из n команд выбирает пару целых чисел A и B ($A \leq B$). Затем Егор проводит K раундов. В каждом раунде он выбирает одно целое число C , и все команды с такой парой чисел (A, B) , что $A \leq C \leq B$, выигрывают в этом раунде. Команда довольна своим выступлением, если она выиграла хотя бы в одном раунде. Помогите Егору выбрать K чисел для раундов так, чтобы сделать наибольшее количество команд довольными.

Например, если в турнире участвуют пять команд, их пары чисел $(1, 2)$, $(1, 4)$, $(3, 6)$, $(4, 7)$, $(5, 6)$, и Егор проводит два раунда, выбирая в первом из них число 2, а во втором — 4, то довольными останутся первые четыре команды. Если же выбрать числа 2 и 5, то все пять команд останутся довольными.

Формат входных данных

Первая строка входных данных содержит два целых числа n и K ($1 \leq n \leq 10\,000, 1 \leq K \leq n, 1 \leq n \cdot K \leq 500\,000$), где n — количество команд, а K — количество раундов в турнире.

Каждая из следующих n строк содержит по два целых числа A и B ($-10^6 \leq A \leq B \leq 10^6$) — пары чисел, которые выбрали команды.

Формат выходных данных

В единственной строке выведите одно целое число — максимальное количество команд, которые могут остаться довольными после турнира.

Примеры

стандартный ввод	стандартный вывод
5 2 1 2 1 4 3 6 4 7 5 6	5
3 2 2 4 1 3 3 5	3
4 1 2 3 1 1 4 5 4 5	2
7 2 5 6 7 9 7 7 1 4 2 3 4 7 4 7	6

Задача J. Последняя булочка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

На очередной вечерке в ГК осталась последняя булочка! Как обычно, чтобы её разыграть, преподаватели устраивают лотерею. Но в этот раз правила совсем другие!

Сначала преподаватели печатают $K!$ билетов: каждый билет содержит K различных номеров от 1 до K , и никакие два билета не идентичны. M из этих билетов выдаются школьникам, претендующим на заветную булочку. Розыгрыш проводится следующим образом: Сначала случайным образом генерируется в общей сложности N различных чисел ($N \geq K$), одно за другим. Если после генерации очередного числа относительный порядок последних K последовательно сгенерированных чисел такой же, как у номеров на любом из выданных билетов, розыгрыш заканчивается немедленно, и счастливый обладатель соответствующего билета выигрывает. Обратите внимание, что жадные преподаватели устроили лотерею так, чтобы был шанс, что булочку не выиграет никто и они смогут съесть её сами.

Например, пусть было распечатано 6 билетов ($K = 3$). Последовательности на билетах: $(1, 2, 3)$, $(1, 3, 2)$, $(2, 1, 3)$, $(2, 3, 1)$, $(3, 1, 2)$ и $(3, 2, 1)$, и среди них выданы $(1, 2, 3)$ и $(1, 3, 2)$ ($M = 2$). Предположим, что сгенерированные $N = 10$ различных случайных чисел такие: $(20, 35, 10, 7, 99, 53, 72, 33, 88, 16)$. Тогда относительный порядок $(7, 99, 53)$, который равен $(1, 3, 2)$, совпадает с выданным билетом $(1, 3, 2)$, поэтому его обладатель выигрывает булочку.

В другом сценарии рассмотрим ситуацию, в которой было распечатано 24 билета ($K = 4$). Сгенерированные последовательности билетов: $(1, 2, 3, 4)$, $(1, 2, 4, 3)$, $(1, 3, 2, 4)$, \dots , $(4, 3, 2, 1)$. Среди них пусть будут выданы $(1, 2, 3, 4)$, $(1, 2, 4, 3)$, $(3, 4, 1, 2)$, $(4, 1, 2, 3)$ и $(4, 2, 3, 1)$ ($M = 5$). Предположим, что сгенерированные $N = 10$ различных случайных чисел такие: $(19, 31, 9, 1, 89, 48, 63, 30, 78, 12)$. Тогда относительный порядок $(89, 48, 63, 30)$, который равен $(4, 2, 3, 1)$, совпадает с выданным билетом $(4, 2, 3, 1)$, поэтому его обладатель выигрывает булочку.

По информации о лотерее, такой как количество распечатанных билетов, набор выданных билетов и последовательность, которая будет сгенерирована, если лотерея не закончится, определите последовательность номеров билета победителя.

Формат входных данных

В первой строке входных данных дано три целых числа K , M и N ($3 \leq K \leq 10\,000$, $1 \leq M \leq \min(K!, 1000)$, $K \leq N \leq 1\,000\,000$, $3 \leq K \cdot M \leq 100\,000$), где K — количество номеров на каждом билете, M — количество выданных билетов, N — длина случайно сгенерированной последовательности для лотереи.

Каждая из следующих M строк содержит K целых чисел, описывающих билет, выданный в раунде.

Последняя строка содержит N различных положительных целых чисел N_i ($1 \leq N_i \leq 100\,000\,000$, $1 \leq i \leq N$), которые представляют собой последовательность сгенерированных чисел для определения победителя.

Формат выходных данных

В единственной строке выведите последовательность чисел в выигрышном билете, разделённых пробелами. Если выигрышного билета нет, выведите 0.

Примеры

стандартный ввод	стандартный вывод
3 2 10 1 2 3 1 3 2 20 35 10 7 99 53 72 33 88 16	1 3 2
4 5 10 1 2 3 4 1 2 4 3 3 4 1 2 4 1 2 3 4 2 3 1 19 31 9 1 89 48 63 30 78 12	4 2 3 1
3 3 7 1 3 2 2 3 1 2 1 3 11 22 33 44 55 66 77	0

Задача К. Лаборатория счастья

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

В ЛКШ открыли секретную лабораторию обработки строк, в которой творятся чудеса. Если у вас есть строка s , можно взять любую её подстроку p длины 1 или 2 и tandemно повторить, то есть вместо p в том же месте написать pp . Например, из «SASHASH» одной такой операцией можно получить «SASH SHASH» и «SA A SHASH». Лаборатория обладает бесконечными мощностями, поэтому таких операций можно сотворить сколько душе угодно.

Издревле известно, что счастье людям приносит строка t . Счастье приносит не только строка t , но и любая строка, содержащая t как подстроку. В ЛКШ строки t нет, в ЛКШ есть строка s длины n и $\frac{1}{2}n(n+1)$ ЛКШ-ат, каждый из которых получит свой отрезок строки s и отправится в секретную лабораторию в поисках счастья. Скольких ЛКШ-ат ждёт успех в поисках счастья?

Формально: из скольки отрезков строки s описанными выше операциями можно породить строку, содержащую t , как подстроку?

Например, если счастье приносит строка $t = SIS$, у строки $s = AISIY$ отрезки AIS, AISI, AISIY, IS, ISI, ISIY, SI, SIY порождают счастье, всего 8 отрезков.

Формат входных данных

В двух строках входных данных даны две строки длины от 1 до 20 000, каждая из которых состоит из заглавных букв латинского алфавита — соответственно s и t .

Формат выходных данных

Выведите одно число — скольких ЛКШ-ат ждёт успех в поисках счастья?

Примеры

стандартный ввод	стандартный вывод
AISIY SIS	8
ATGTG TTG	9
CACTGT CCTTG	6
PQRPQR PQR	7
BCDBCD BCDBCD	1

Задача L. Бегущий город

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

Саша и Вова придумывают задания на Бегущий город в Судиславле. Как известно, Судиславль представляет собой n перекрестков и m двусторонних дорог. Между каждой парой перекрестков существует не более одной дороги. Гарантируется, что между каждой парой перекрестков существует путь. Для каждого перекрестка i Саша придумал вопрос типа a_i . Вова хочет найти простой путь по Судиславлю, который удовлетворяет следующим условиям:

- Пусть P — последовательность типов вопросов на перекрестках вдоль пути, перечисленных в порядке их посещения.
- Если P является неубывающей, то *интересность* этого пути равна количеству различных типов вопросов на пути.
- Иначе *интересность* этого пути равна 0.

Помогите найти Саше и Вове путь с максимальной *интересностью*.

Напомним, что простой путь — это путь, который проходит через каждый перекресток не более одного раза.

Формат входных данных

В первой строке дано два натуральных числа n и m — количество перекрестков и количество дорог в Судиславле ($2 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 2 \cdot 10^5$).

Во второй строке дано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \cdot 10^5$) — тип вопроса на i -м перекрестке.

В следующих m строках описаны дороги. Дорога с номером i описана парой чисел v_i и u_i — номера перекрестков ($1 \leq v_i \leq u_i \leq n$).

Формат выходных данных

Выведите максимальную возможную интересность пути.

Примеры

стандартный ввод	стандартный вывод
5 6 10 20 30 40 50 1 2 1 3 2 5 3 4 3 5 4 5	4
4 5 1 10 11 4 1 2 1 3 2 3 2 4 3 4	0
10 12 1 2 3 3 4 4 4 6 5 7 1 3 2 9 3 4 5 6 1 2 8 9 4 5 8 10 7 10 4 6 2 8 6 7	5

Задача М. Чиселковые методы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Параллель ML очень любит чиселки. А ещё больше она любит замечать чиселковые свойства чиселок.

Например, пусть дано целое чиселко $a = \overline{a_1 a_2 \dots a_n}$ из n цифр и положительное целое чиселко k . a называется k -чиселковым, если произведение его цифр, которое равно $a_1 \cdot a_2 \cdot \dots \cdot a_n$, делится на k .

Заметим, что чиселко 0 делится на любое целое положительное чиселко.

Например, если $a = 2349$ и $k = 12$, то произведение цифр чиселка a , $2 \cdot 3 \cdot 4 \cdot 9 = 216$ делится на $k = 12$, то есть 2349 является 12-чиселковым. Если $a = 2349$ и $k = 16$, то произведение цифр a , $2 \cdot 3 \cdot 4 \cdot 9 = 216$ не делится на 16, поэтому 2349 не 16-чиселковое.

Для трёх целых чиселок k, L, R найдите $x \bmod 998\,244\,353$, где x — количество k -чиселковых чиселок в диапазоне $[L, R]$.

Формат входных данных

В единственной строке входных данных даны три целых чиселка k, L и R ($1 \leq k \leq 10^{17}$, $1 \leq L \leq R \leq 10^{20}$).

Формат выходных данных

В единственной строке выведите $x \bmod 998\,244\,353$, где x — количество k -чиселковых чиселок среди чисел в диапазоне $[L, R]$, где L и R включены в диапазон.

Примеры

стандартный ввод	стандартный вывод
5 1 20	4
5 50 100	19
15 11 19	0

