

Задача А. Календарь спецкурсов

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

Андрей Сергеевич решил распланировать спецкурсы на следующие дни. Он взял свой квадратный календарь, в котором n строк и n столбцов, и выписал интересность распределённых курсов в соответствующие ячейки календаря.

Обозначим за $a_{i,j}$ интересность спецкурса в i -й строке j -м столбце календаря.

Он считает день на пересечении i -й строки и j -го столбца *сбалансированным*, если сумма интересностей в i -й строке отличается от суммы интересностей в j -м столбце не более чем на $a_{i,j}$.

Ему стало интересно, сколько существует *сбалансированных* дней в составленном распределении. Помогите ему и скажите это число.

Формат входных данных

В первой строке дано одно число n — длина стороны календаря ($1 \leq n \leq 1000$).

В последующих n строках дано по n неотрицательных чисел — интересность спецкурса в соответствующие дни ($a_{i,j} \leq 10^5$).

Формат выходных данных

Выведите количество *сбалансированных* дней.

Примеры

стандартный ввод	стандартный вывод
2 1 9 0 5	1
4 1 1 4 3 1 2 9 2 1 4 2 1 5 2 1 3	10
4 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 0	7

Замечание

Рассмотрим первый пример:

Сумма в первой строке равна $1 + 9 = 10$, сумма во второй строке равна $0 + 5 = 5$. Сумма в первом столбце равна $1 + 0 = 1$, сумма во втором столбце равна $9 + 5 = 14$.

Единственным интересным днём является день на пересечении первой строки и второго столбца, так как $|14 - 10| \leq 9$.

Остальные дни интересными не являются: $|10 - 1| > 1$, $|5 - 1| > 0$, $|14 - 5| > 5$.

Задача В. Погода на хуторе

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

Смена ЛКШ.2077 будет идти n дней. В i -й день температура на хуторе будет принимать все значения на отрезке от l_i до r_i . Главный метеоролог ЛКШ Денис Павлович считает, что смена пройдет удачно, если в каждый из дней можно выбрать температуру t_i такую, что $l_i \leq t_i \leq r_i$ и сумма выбранных температур за все дни равна 0. Так как Денис Павлович занят отбоем, определить успешность смены предстоит вам.

Формат входных данных

В первой строке дано одно натуральное число n — количество дней в смене ($1 \leq n \leq 2 \cdot 10^5$).

В следующих n строках описаны дни. В $i + 1$ строке описан i -й день числами l_i и r_i ($-10^9 \leq l_i \leq r_i \leq 10^9$).

Формат выходных данных

Выведите «Yes», если в каждый из дней можно выбрать температуру из данного промежутка, чтобы их сумма была равна 0, и «No» иначе.

Примеры

стандартный ввод	стандартный вывод
3 3 5 -4 1 -2 3	Yes
3 1 2 1 2 1 2	No
6 -87 12 -60 -54 2 38 -76 6 87 96 -17 38	Yes

Замечание

В первом примере можно выбрать $t_1 = 3.5$, $t_2 = -3.1$, $t_3 = -0.4$.

Задача С. Тайные собрания

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В ЛКШ, как вам всем известно, есть жесткий отбой, во время которого преподады очень жестко следят за соблюдением определенных правил (выключен свет, комната не производит громких звуков).

ЛКШ можно представить как граф, в котором есть N домиков, между которыми существует M дорожек, дорожки бывают разных длин, длина i -й дорожки равна l_i . Также известно, что в первых C домиках живут преподады. Преподады ночью довольно ленивые, их лень характеризуется числом L . Если преподад живет в домике i , то ночью он может проверить все домики, расстояние до которых от домика i не превосходит L .

Недавно в ЛКШ вернули систему объяснительных. Но преподаватели, как правило, лояльно относятся к школьникам, поэтому они заставляют домик писать объяснительные, если он привлек внимание как минимум K преподавателей.

Тайный совет школьников интересуется, в каких домиках небезопасно проводить собрания. Любое собрание тайного совета производит много шума. Считается, что в домике небезопасно проводить собрание, если это приведет к написанию объяснительных, то есть шум в домике привлечет не менее K преподавателей. Совет поручил вам найти все такие домики, при этом их не интересуют преподавательские домики, их сообщать не надо.

Формат входных данных

Первая строка содержит пять чисел, разделенных пробелами N, M, C, L, K — количество домиков, дорожек, домиков преподадов, ленивость преподадов и количество преподавателей, внимание которых нужно привлечь, чтобы писать объяснительную ($1 \leq C \leq N \leq 5 \cdot 10^4$, $1 \leq M \leq 10^5$, $1 \leq L \leq 10^9$, $1 \leq K \leq 10$).

Каждая из последующих M строк содержит по 3 числа: u_i, v_i, l_i ($1 \leq u_i, v_i \leq N$, $1 \leq l_i \leq 10^9$, $v_i \neq u_i$) — номера домиков и длина соединяющей их дорожки.

Формат выходных данных

В первой строке выведите количество небезопасных домиков. В последующих строках выведите номера домиков по одному в каждой строке.

Примеры

стандартный ввод	стандартный вывод
3 3 1 4 1 1 2 3 1 3 5 2 3 2	1 2
4 3 2 101 2 1 2 1 2 3 100 1 4 10	2 3 4
4 3 2 100 2 1 2 1 2 3 100 1 4 10	1 4

Замечание

В первом примере есть преподский домик с номером 1. Преподаватель от него может прийти только до домика 2, поскольку он находится на расстоянии 3 от домика 1. Поэтому только 2-й является небезопасным домиком.

Во втором примере мы имеем два преподских домика с номерами 1 и 2, и два домика школьников с номерами 3 и 4 на расстоянии 101 от обоих преподских домиков. Поэтому оба домика 3 и 4 являются небезопасными.

Задача D. Рептилоиды атакуют хутор

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

Вы живёте на хуторе. Однажды ночью вас разбудили странные звуки. Вы выглянули в окно и обнаружили n рептилоидов, выстроившихся в очередь перед дыркой в заборе с явными намерениями пробраться на территорию ЛКШ. Над головой каждого рептилоида высвечивается его *ранг*. *Ранг* первого (ближайшего к забору) рептилоида равен a_1 , *ранг* второго равен a_2 , и так далее, *ранг* последнего равен a_n .

Вам безумно интересно, что же задумали эти рептилоиды, но скоро подъём и зарядка, поэтому у вас ровно m минут, чтобы понаблюдать за ними. Каждую минуту будет происходить какое-то одно событие.

События бывают следующих видов:

- $+ x$ — новый рептилоид с *рангом* x встаёт в **конец** очереди;
- $-$ — **первый** рептилоид в очереди покидает её;
- r — рептилоиды в очереди совершают *ритуал*;
- $i x$ — *ранг всех* рептилоидов в очереди **увеличивается** на x ;
- $d x$ — *ранг всех* рептилоидов в очереди **уменьшается** на x ;
- $?$ — вывести *могущество* очереди рептилоидов.

Во время *ритуала* первый рептилоид меняется местом с последним, второй с предпоследним и т.д. Другими словами, очередь переворачивается. Например, очередь $\{5, 8, 1, 9, 2\}$ после *ритуала* превратится в очередь $\{2, 9, 1, 8, 5\}$.

Могущество очереди рептилоидов это *ранг* первого рептилоида плюс удвоенный *ранг* второго плюс утроенный *ранг* третьего и т.д. Другими словами, это значение $\sum_{i=1}^n i \cdot a_i$. Например, *могущество* очереди $\{4, 1, 2\}$ равно $1 \cdot 4 + 2 \cdot 2 + 3 \cdot 2 = 11$.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество рептилоидов в очереди.

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$) — *ранги* рептилоидов в очереди.

В третьей строке дано целое число m ($1 \leq m \leq 2 \cdot 10^5$) — количество событий.

В следующих m строках описаны события.

Каждое событие задаётся символом c («+», «-», «r», «i», «d» или «?») — типом события.

Для событий с типами «+», «i» и «d» задаётся параметр x ($1 \leq x \leq 10^4$).

Формат выходных данных

Для каждого запроса типа «?» выведите в отдельной строке *могущество* очереди рептилоидов.

Пример

стандартный ввод	стандартный вывод
5	77
1 4 5 2 9	76
10	68
?	143
+ 4	113
-	
?	
r	
?	
i 5	
?	
d 2	
?	

Замечание



Задача Е. Горилла и AVX

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

Однажды огромный самец гориллы сходил на спецкурсе по AVX. Обрадовавшись возможности быстро выполнять программы с квадратичной сложностью, он написал следующий код:

```
mod = 998244353
sum = 0
for i in range(n):
    for j in range(i + 1, n):
        sum += int(str(a[i]) + str(a[j]))

print(sum % mod)
```

Формализуем задачу:

Дан массив a из n целых чисел.

Введём функцию $f(x, y)$ — число, полученное выписыванием десятичных представлений чисел x и y подряд друг за другом. Например, $f(12, 54) = 1254$, $f(82, 1) = 821$.

Требуется посчитать сумму: $\left(\sum_{i=1}^{n-1} \sum_{j=i+1}^n f(a_i, a_j) \right) \bmod 998244353$.

К сожалению, у гориллы не получилось придумать, как в этой задаче применять AVX, и он обратился к вам за помощью. Помогите горилле написать более оптимальное решение.

Формат входных данных

В первой строке входного файла даётся одно целое число N — длина массива ($2 \leq N \leq 2 \cdot 10^5$).

Во второй строке даются через пробел числа a_i ($1 \leq a_i \leq 10^9$).

Формат выходных данных

В единственной строке выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 3 14 15	2044
5 1001 5 1000000 1000000000 100000	625549048

Замечание

В первом примере из условия сумма равна:

$((f(3, 14) + f(3, 15) + f(14, 15)) \bmod 998244353) = ((314 + 315 + 1415) \bmod 998244353) = (2044 \bmod 998244353) = 2044$

Задача F. Вечёрка

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

Неожиданно преподаватели сообщили школьникам, что со следующего дня изменится уже всем привычная выдача вечёрки. Теперь школьники забирают её сами! Выдача будет работать по следующим правилам:

Ответственные за вечёрку выделяют квадратную область, в которой ровно $(n+1) \times (n+1)$ клеток. За (i, j) обозначают клетку на пересечении i -й строки сверху вниз и j -го столбца слева направо.

В клетках с номерами (i, j) , где $1 \leq i, j \leq n$ находятся стрелки. Они бывают двух видов: R – стрелка направо и D – стрелка вниз.

В клетках с номерами (i, j) , где $i = n+1$ или $j = n+1$ находятся столы с вечёркой. Для каждого стола известно, сколько булочек будет выдано школьнику, который придёт в клетку (i, j) . При этом стола $(n+1, n+1)$ не существует.

Перед началом выдачи вечёрки на каждой клетке стоит школьник. Ровно в 18:30 они начинают двигаться к столам, переходя по стрелкам. И так до тех пор, пока ЛКШата не дойдут до них.

Всего будет q изменений. Ответственные за вечёрку пока не уверены в направлении стрелок, и поэтому в конце каждого дня они меняют направление в клетке (i, j) . После каждого изменения преподаватели хотят понять, сколько булочек будет выдано на вечёрке.

Формат входных данных

Первая строка содержит одно число n ($1 \leq n \leq 1500$).

Далее даются n строк. В i -й из них даётся строка из n символов, каждый из которых равен R или D, за которой через пробел следует количество выдаваемых в секторе $(i, n+1)$ булочек.

$n+2$ строка содержит n чисел, j -е из которых равно количеству выдаваемых булочек в клетке $(n+1, j)$.

Следующая строка содержит одно число q ($1 \leq q \leq 1500$).

Затем следует q строк, каждая из которых содержит по два числа i и j , которые обозначают, в какой строке и каком столбце меняется указатель ($1 \leq i, j \leq n$).

Гарантируется, что за каждым столом выдаётся не менее 1 булочки и не более 500 булочек.

Формат выходных данных

Выведите $q+1$ число: количество выданных булочек в начальном состоянии, а также количество выданных булочек после каждого изменения.

Пример

стандартный ввод	стандартный вывод
2	62
RR 1	71
DD 10	62
10 50	71
4	151
1 1	
1 1	
1 1	
2 1	

Замечание

Изначально в клетку (1, 3) придут два школьника, в клетки (3, 1) и (3, 2) – по одному, в клетку (2, 3) не придёт никто.

Итоговая сумма $1 \cdot 2 + 10 \cdot 1 + 50 \cdot 1 = 62$.

После первого изменения указатели будут выглядеть следующим образом:

DR

DD

После этого в клетки (1, 3) и (3, 2) придёт по одному школьнику, в клетку (3, 1) придут два школьника, в клетку (2, 3) не придёт никто.

Итоговая сумма: $1 \cdot 1 + 10 \cdot 2 + 50 \cdot 1 = 71$.

Задача G. Эпидемия в ЛКШ

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

Как всем известно, школьники в ЛКШ — порядочные люди, поэтому они все приходят на линейку. Школьники любят стабильность, поэтому каждый школьник встаёт на одно и то же место каждый день.

Площадь на которой проходит линейка можно представить в виде бесконечной клетчатой сетки, в каждой клетке которой стоит школьник, то есть школьников бесконечное количество.

Оказывается, один из школьников был больным, из-за чего в ЛКШ началась эпидемия. Каждый день болеющий школьник заражает другого школьника, если тот находится рядом с ним и ещё не заразился, после болезни появляется иммунитет. Клетки считаются соседними, если соприкасаются по стороне, то есть у каждой клетки ровно четыре соседа.

Заболевший школьник приходит к Рите в медпункт на следующий день после линейки, как он заразился, так как симптомы проявляются не сразу. И его вылечивают в этот же день.

Первый школьник обратился в медпункт после линейки и это считается первый день эпидемии.

Помогите Рите узнать, сколько школьников придут в k -й день к ней в медпункт.

Формат входных данных

Первая и единственная строка содержит целое число k ($1 \leq k \leq 10^8$) — номер дня с начала эпидемии в ЛКШ.

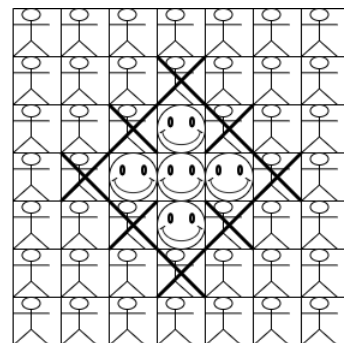
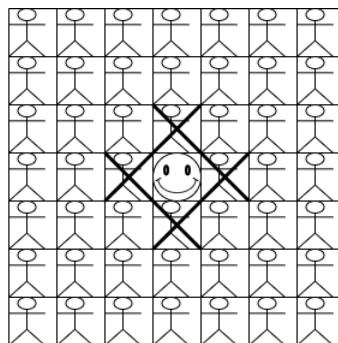
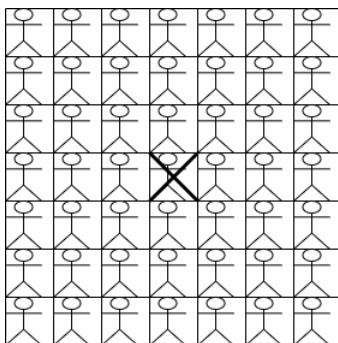
Формат выходных данных

Выведите единственное число — количество людей пришедших в медпункт в k -й день ЛКШ.

Примеры

стандартный ввод	стандартный вывод
1	1
2	4
3	8

Замечание



Первые три дня эпидемии

Задача Н. Бегущий город

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
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 максимально 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

Задача I. Арсений и карты

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

На столе перед Арсением в ряд лежат карты a_1, a_2, \dots, a_n . Чтобы перемешать их, он **ровно один** раз выбирает отрезок $[l, r]$ ($1 \leq l \leq r \leq n$) и сортирует карты на этом отрезке по **неубыванию**. Например, если перед Арсением лежат карты $\{8, 1, 5, 3, 2, 2, 9\}$, он может выбрать отрезок $[2, 5]$ и получить последовательность $\{8, 1, 2, 3, 5, 2, 9\}$.

Сможет ли Арсений получить заданную последовательность b_1, b_2, \dots, b_n с помощью **ровно одного** перемешивания?

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество карт.

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — последовательность карт на столе.

В третьей строке даны n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^9$) — последовательность карт, которую хочет получить Арсений.

Формат выходных данных

Выведите «YES», если Арсений может получить заданную последовательность карт, иначе выведите «NO».

Примеры

стандартный ввод	стандартный вывод
5 1 4 2 2 4 1 4 4 2 2	NO
6 5 1 2 5 3 5 5 1 2 3 5 5	YES
3 4 1 2 1 4 7	NO
1 7 7	YES
7 4 4 1 7 5 3 8 4 1 4 5 7 3 8	YES

Задача J. 20:31. Прибыл сдавать документы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Вася ездит в ЛКШ каждый год. И этот — не исключение. Однако самым интересным челленджем он считает сдачу документов в начале смены. Как известно, это тонко организованный процесс, и для того, чтобы успеть сдать достаточное количество документов, надо правильно распределить своё время, учитывая информацию о времени закрытия каждой станции приёма определённого документа.

Известно, что есть N станций по сбору документов (на каждой станции собирают уникальный документ). Станции пронумерованы от 1 до N . У i -й станции есть время e_i , когда она закрывается. Вася приезжает в момент времени A и хочет посетить i -ю станцию в момент времени $A + t_i$. Чтобы успеть сдать документ, Вася должен прийти на станцию строго раньше, чем момент e_i , когда она закроется. Время, затрачиваемое на перемещение между станциями и сдачу, считается равным нулю.

У Васи есть Q запросов. Каждый запрос выглядит как предполагаемое время прибытия Васи A и число станций C , которые он хочет успеть посетить за сегодня. Для каждого запроса сообщите, сможет ли он посетить не менее чем C станций, если его время прибытия будет равно A .

Формат входных данных

Первая строка содержит числа N и Q ($1 \leq N, Q \leq 2 \cdot 10^5$).

Во второй строке даются через пробел числа $e_1, e_2, e_3, \dots, e_N$ ($1 \leq e_i \leq 10^6$) — времена закрытий станций.

В третьей строке через пробел даются числа $t_1, t_2, t_3, \dots, t_N$ ($1 \leq t_i \leq 10^6$) — задержка между приездом и посещением i -й станции.

Каждая из следующих Q строк содержит по два целых числа ($1 \leq C \leq N$) и A ($1 \leq A \leq 10^6$) — количество станций, которое хочет успеть посетить Вася и время его прибытия.

Формат выходных данных

Для каждого из Q запросов выведите «YES» или «NO» на новой строке.

Пример

стандартный ввод	стандартный вывод
5 5	YES
3 5 7 9 12	NO
4 2 3 3 8	YES
1 5	YES
1 6	NO
3 3	
4 2	
5 1	

Замечание

Рассмотрим первые два запроса из теста выше:

Первый запрос: $C = 1, A = 5$, следовательно выпишем $t_i + A$ и e_i

- $i = 1 : 4 + 5 \geq 3$
- $i = 2 : 2 + 5 \geq 5$
- $i = 3 : 3 + 5 \geq 7$
- $i = 4 : 3 + 5 < 9$ — Вася успеет
- $i = 5 : 8 + 5 \geq 12$

Значит, Вася успеет сдать $C = 1$ документов.

Второй запрос: $C = 1, A = 6$, следовательно выпишем $t_i + A$ и e_i

- $i = 1 : 4 + 6 \geq 3$
- $i = 2 : 2 + 6 \geq 5$
- $i = 3 : 3 + 6 \geq 7$
- $i = 4 : 3 + 6 \geq 9$ — Вася уже не успеет сдать документ
- $i = 5 : 8 + 6 \geq 12$

Значит, Вася не успеет сдать $C = 1$ документов.

Задача К. Экзистенция, как она есть

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

Велоклуб — мероприятие философское. Многие философы-экзистенциалисты замечали, что длинные поездки способствуют познанию сути мира. Андрей, вдохновившись книгами, решил прокатиться по Берендеевым полянам.

Во время первой поездки он понял, в чём суть этого мира. А именно в том, чтобы доехать до конечной точки пути и не умереть от усталости.

Забавно, что Берендеевы поляны можно представить в виде ориентированного графа на N вершинах, где ребра это дороги, а вершины — перевалочные пункты. Кроме того, для каждой дороги известно:

- b_i — красота i -й дороги
- s_i — сколько выносливости отнимает проезд по i -й дороге
- любая дорога ведёт из контрольного пункта с меньшим номером в контрольный пункт с большим номером

Андрей хочет доехать от 1-й вершины до N -й вершины, и при этом получить максимальное удовольствие от поездки. Удовольствие он оценивает как сумму красот, делённую на сумму усталостей на дорогах, по которым он проехал.

Помогите Андрею оценить максимальное удовольствие, которое он может получить во время своей поездки.

Формат входных данных

В первой строке через пробел даны два числа N и M ($2 \leq N \leq 2 \cdot 10^5$, $1 \leq M \leq 2 \cdot 10^5$).

В следующих M строках описаны дороги. В i -й строке задано описание i -й дороги: через пробел заданы четыре целых числа u_i , v_i , b_i , s_i ($1 \leq u_i < v_i \leq N$, $1 \leq b_i, s_i \leq 10^4$), где u_i — откуда идёт дорога, v_i — куда ведёт дорога, b_i - красота дороги, s_i - усталость от дороги.

Андрей гарантирует Вам, что существует хотя бы один путь от 1-й до N -й вершины.

Формат выходных данных

Выведите число с точностью 10^{-9} , которое будет характеризовать максимальное удовольствие, которое может получить от поездки Андрей.

Примеры

стандартный ввод	стандартный вывод
5 7 1 2 3 6 1 3 9 5 2 3 1 5 2 4 5 3 2 5 1 9 3 4 4 8 4 5 2 7	0.7500000000
3 3 1 3 1 1 1 3 2 1 1 3 3 1	3.0000000000

Задача L. Суперпростые числа

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

Для простого числа p введём функцию $num(p)$, значение которой равно номеру этого простого числа. Например, $num(5) = 3$.

Для натурального числа x введём функцию $pr(x)$, значение которой равно x -ому простому числу. Например, $pr(3) = 5$.

Для натурального числа n введём функцию $rev(n)$. Значение функции $rev(n)$ равно числу, полученному из n разворотом его десятичной записи. Например, $rev(34) = 43$.

Для натурального числа m введём функцию $\prod(m)$, значение которой равно произведению цифр десятичной записи числа m . Например, $\prod(24) = 2 \cdot 4 = 8$, $\prod(1024) = 1 \cdot 0 \cdot 2 \cdot 4 = 0$.

Назовём *суперпростым* простое число p , удовлетворяющее двум свойствам:

- $pr(rev(num(p))) = rev(p)$. Например, для $p = 73$: $num(73) = 21$, $rev(num(p)) = rev(21) = 12$, $pr(rev(num(p))) = pr(12) = 37 = rev(73)$.
- $\prod(p) = num(p)$. Например, для $p = 73$: $num(73) = 21$, $\prod(73) = 7 \cdot 3 = 21 = num(73)$.

Формат входных данных

В единственной строке находится одно целое число k ($1 \leq k \leq 10$).

Формат выходных данных

Выведите k -е *суперпростое* число (число, удовлетворяющее двум описанным свойствам). Если k -го *суперпростого* числа не существует, выведите -1.

Пример

стандартный ввод	стандартный вывод
1	73