

## Задача А. Отгадай-ка

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

Вадим загадал два целых числа  $X$  и  $Y$ , однако он сообщил вам лишь их сумму  $A = X + Y$  и их разность  $B = X - Y$ . Вам кажется, что он мог вас обмануть. Проверьте, существуют ли подходящие  $X$  и  $Y$ .

### Формат входных данных

В первой строке даётся целое число  $A$  — сумма  $X$  и  $Y$  ( $-10^{100} \leq A \leq 10^{100}$ ).

Во второй строке даётся целое число  $B$  — разность  $X$  и  $Y$  ( $-10^{100} \leq B \leq 10^{100}$ ).

### Формат выходных данных

Выведите «*Correct*», если подходящие  $X$  и  $Y$  существуют, в обратном случае выведите «*Liar*». Выводить кавычки не нужно.

### Примеры

стандартный ввод	стандартный вывод
12 2	Correct
1 0	Liar

### Замечание

В первом примере, если  $X = 7$  и  $Y = 5$ , то их сумма равна 12, а их разность равна 2.

Во втором примере подходящих  $X$  и  $Y$  не существует, так как их разность равна 0, а это значит, что  $X = Y$ , но сумма двух равных целых чисел никогда не может равняться 1.

## Задача В. Сделай 99

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

На странице одной книги Вадим нашёл старую добрую головоломку. Она заключается в том, что в ней надо вставить знаки сложения в число так, чтобы получить другое число. В этой головоломке дана строка:

**987654321**

Путём вставки знаков сложения в эту строку необходимо получить ровно 99. Вадим быстро нашёл решение этой задачи, а сможете ли вы?

### Формат входных данных

В этой задаче ровно один тест. Для вашего удобства во входных данных даётся единственная строка «987654321» (без кавычек). Считывать её необязательно.

### Формат выходных данных

Выведите единственную строку без пробелов — математическое выражение, требуемое в условии. Оно должно состоять из цифр 9, 8, 7, 6, 5, 4, 3, 2, 1 в таком порядке и операций + (код 43). Выражение должно быть математически корректным: операции сложения должны стоять строго между операндами. Разрешено ставить несколько цифр рядом, чтобы образовать многозначное число. При выполнении всех операций выражение должно давать 99.

### Пример

стандартный ввод	стандартный вывод
987654321	987+65+432+1

### Замечание

Обратите внимание, что ответ в примере неверен и приведён только для иллюстрации формата. В нём арифметическое выражение соответствует формату, но результат равен 1485, а не 99.

## Задача С. Детский сад

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

В детском саду ЧП — дети нашли и съели годовой запас конфет! Сейчас перед нянечкой стоит непростая задача — понять, кто съел больше всех конфет, чтобы оказать ему помощь в первую очередь. Однако дети очень боятся наказания, поэтому наотрез отказываются говорить, сколько конфет съели. Единственное, что готов сообщить  $i$ -й ребёнок, — это число  $a_i$  — сколько съели все остальные дети, кроме него. В этом возрасте дети уже достаточно хорошо складывают и вычитают числа, поэтому вы можете считать, что эти сведения верны. Также точно известно, что ни один ребёнок не мог съесть больше 100 конфет. Помогите нянечке понять, кто из детей съел больше всех конфет, или сообщите, что это невозможно установить однозначно.

### Формат входных данных

Первая строка содержит единственное целое число  $N$  — общее количество детей ( $2 \leq N \leq 100$ ).

Следующие  $N$  строк содержат по одному целому числу  $a_1, a_2, \dots, a_N$  — числа, названные каждым ребёнком ( $1 \leq a_i \leq 100 \cdot (N - 1)$ ).

Гарантируется, что существуют такие количества съеденных конфет, что все названные детьми числа верны, при этом каждый ребёнок съел не меньше одной и не больше ста конфет.

### Формат выходных данных

Если невозможно однозначно понять, кто съел больше всех конфет, то в единственной строке выведите одно число  $-1$ . В противном случае выведите одно число — индекс ребёнка, съевшего больше всех конфет.

### Примеры

стандартный ввод	стандартный вывод
3 38 38 36	3
3 36 36 36	-1

### Замечание

В первом примере первый и второй ребёнок съели по 18 конфет, а третий 20.

Во втором примере каждый из детей съел по 18 конфет, так что невозможно сказать, кто съел больше всех.

## Задача D. Саморезы в доме

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

Дома у Вадима почти закончился ремонт, и сейчас ему осталось лишь полностью закрутить  $N$  саморезов в одну доску. Эти саморезы расположены в один ряд, поэтому они пронумерованы числами от 1 до  $N$ ,  $i$ -й саморез недокручен на  $l_i$  сантиметров.

У Вадима в арсенале есть отвёртка и шуруповёрт. С помощью отвёртки он может закрутить любой саморез на 1 сантиметр за 1 секунду. А с помощью шуруповёрта Вадим может закрутить полностью любой саморез за 1 секунду вне зависимости от того, на сколько он недокручен. Однако же шуруповёрт частично разряжен — у него осталось заряда на закрутку лишь  $K$  саморезов.

Ремонт уже давно пора заканчивать, поэтому подскажите Вадиму, какое наименьшее количество секунд у него может уйти на полную закрутку всех саморезов.

### Формат входных данных

В первой строке даны два целых числа  $N$  и  $K$  — количество саморезов и заряд шуруповёрта ( $1 \leq N, K \leq 100$ ).

Во второй строке даны  $N$  целых чисел  $l_i$  — описание недокрученности  $i$ -го самореза ( $1 \leq l_i \leq 10^9$ ).

### Формат выходных данных

Выведите наименьшее количество секунд, которое может уйти у Вадима на полную закрутку всех саморезов.

### Примеры

стандартный ввод	стандартный вывод
4 1 2 12 4 9	16
3 10 16 15 12	3

### Замечание

В первом примере Вадим может закрутить шуруповёртом 2-й саморез за 1 секунду, а три остальных — с помощью отвёртки. Суммарно у него это займёт 16 секунд.

Во втором примере Вадим может закрутить все саморезы с помощью шуруповёрта за 3 секунды.

## Задача Е. Подстольный теннис

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

А вы уже слышали про давно набирающий популярность подстольный теннис? Эта игра была придумана Вадимом и заключается в том, что два игрока ракетками отбивают мячик под специальным столом до тех пор, пока у одного из них мячик не вылетит из-под стола.

Как всем известно, ракетки в подстольном теннисе имеют интересный вид. Все ракетки симметричные, то есть их можно отразить относительно вертикальной оси. У каждой ракетки есть верхушка, причём толщина верхушки либо 2 см, либо 4 см. Затем идёт  $N$  секций, причём каждая из них с определённой неровностью. Эти неровности могут перевести ракетку из толщины 2 см между секциями в толщину 4 см между двумя следующими, могут перевести в обратную сторону или могут не переводить вовсе. После всех секций идёт рукоятка, которая также может быть толщиной либо 2 см, либо 4 см. Толщина верхушки зависит от неровности самой первой секции сверху, а толщина рукоятки зависит от неровности самой первой секции снизу.

Высотой ракетки называют количество секций. Ниже приведены примеры ракеток высоты 3:

```
----  --  ----  --  
 \ /  / \  ) (  ( )  
( )  ) (  \ /  / \  
/ \  \ /  ( )  ) (  
IIII  II  II  IIII
```

Чтобы игра была *великолепной*, ракетки оппонентов должны быть такими, чтобы их можно было сложить рядом, перевернув одну из них. То есть неровность каждой секции одной ракетки должна подходить к неровности соответствующей секции другой ракетки, а именно первая к последней, вторая к предпоследней и так далее. Ниже приведены примеры совмещающихся ракеток высоты 3:

```
---- II  IIII --  
 \ // \ ) (( )  
( )) ( \ // \  
/ \ \_/_/ (__) )  
IIII  IIII
```

Вадим предложил вам сыграть в подстольный теннис и показал свою ракетку. Подберите себе такую ракетку, чтобы ваша игра против Вадима была великолепной.

### Формат входных данных

В первой строке даётся целое число  $N$  — высота ракетки ( $1 \leq N \leq 10^5$ ).

Во второй строке описывается верхушка ракетки одной из следующих двух строк:

- «. \_ \_ .» — толщина верхушки 2 см
- «\_ \_ \_ \_» — толщина верхушки 4 см

В следующих  $N$  строках описываются неровности секций ракетки одной из следующих четырёх строк:

- «( . )» — секция с неровностью, которая оставляет толщину ракетки 2 см
- «) . . (» — секция с неровностью, которая оставляет толщину ракетки 4 см
- «/ . . \» — секция с неровностью, у которой толщина ракетки сверху 2 см, снизу 4 см
- «\ . . /» — секция с неровностью, у которой толщина ракетки сверху 4 см, снизу 2 см

В следующей строке ввода описывается рукоятка ракетки одной из следующих двух строк:

- «.II.» — толщина рукоятки 2 см
- «IIII» — толщина рукоятки 4 см

Гарантируется, что верхушка подобрана корректно относительно первой секции. Гарантируется, что любые две соседние секции имеют одинаковую толщину между ними. Гарантируется, что рукоятка подобрана корректно относительно  $N$ -й секции.

### Формат выходных данных

Выведите такую ракетку, чтобы игра против ракетки, данной во вводе, была великолепной. Выведите ракетку в описанном выше формате.

### Пример

стандартный ввод	стандартный вывод
7	----
.---	)..(
(..)	\../
/..\	(..)
\../	/..\
/..\	\../
)..(	/..\
\../	)..(
(..)	IIII
.II.	

### Замечание

Заметьте, что при вводе и выводе ракеток используются точки. Ниже приведены коды символов, использующиеся в задаче:

- «\_» — код символа 95
- «(» — код символа 40
- «>» — код символа 41
- «/» — код символа 47
- «\» — код символа 92
- «I» — код символа 73
- «.» — код символа 46

Чтобы вывести символ «\» используйте «\\».

## Задача F. Торжественные массивы

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

Сегодня в квалификационном туре некоторые команды борются не только за выход на четвертьфинал ICPC, но и стремятся занять призовые места в юбилейном 30-м командном чемпионате УрФУ по программированию. Жюри долго размышляло, как поздравить участников. В итоге единогласно было решено подарить каждой команде массив, причём не обычный, а торжественный. Массив  $a$  длины  $N$  называется торжественным, если:

- $0 \leq a_i < K$  для всех  $i$  от 1 до  $N$
- сумма чисел в массиве  $(a_1 + a_2 + \dots + a_N)$  нацело делится на  $K$

Чтобы никого не обидеть, жюри хочет подарить всем командам уникальные массивы одинаковой длины и с одним и тем же  $K$ , однако если сделать неверный выбор, то таких массивов может не хватить. В итоге жюри сошлись во мнении, что для идеального подарка нужна программа, которая по данным  $N$  и  $K$  считает количество различных торжественных массивов. Помогите жюри и напишите такую программу. Поскольку ответ может быть слишком большим, выведите его остаток от деления по модулю  $10^9 + 7$ .

### Формат входных данных

В единственной строке даны два целых числа  $N$  и  $K$  — параметры торжественного массива ( $1 \leq N \leq 10^5, 1 \leq K \leq 10^9$ ).

### Формат выходных данных

Выведите количество различных торжественных массивов по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
3 1	1
3 2	4
2025 30	531288623

### Замечание

В первом примере существует только один торжественный массив:  $[0, 0, 0]$ .

Во втором примере существует 4 торжественных массива:  $[0, 0, 0]$ ,  $[1, 0, 1]$ ,  $[1, 1, 0]$ ,  $[0, 1, 1]$ .

## Задача G. Шахматный турнир

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

Тренер Олег решил организовать в шахматной школе турнир по круговой системе, однако он не очень хорошо разобрался, что это значит, поэтому сделал следующим образом:  $N$  детей встают в круг, затем каждый играет одну партию со своим левым соседом и одну партию с правым. Точно известно, что дети играют очень упорно, поэтому каждая партия заканчивается победой одного из участников, ничьих быть не может.

Отлучившись и вернувшись через 5 минут, Олег обнаружил, что все партии уже сыграны, а каждый ребенок хвастается своим числом побед.  $i$ -й ребёнок утверждает, что выиграл ровно  $a_i$  партий. Олег подозревает, что кто-то из детей мог соврать. Помогите ему понять, являются ли утверждения детей непротиворечивыми, то есть существуют ли такие исходы для каждой партии, которые приводят к заявленному количеству побед у каждого ребёнка.

### Формат входных данных

Первая строка содержит единственное целое число  $N$  — общее количество детей ( $3 \leq N \leq 10^5$ ).

Вторая строка содержит  $N$  целых чисел  $a_1, a_2, \dots, a_N$  — утверждения  $i$ -го ребёнка ( $0 \leq a_i \leq 2$ ).

### Формат выходных данных

В единственной строке выведите «YES», если существуют такие исходы партий, при которых все утверждения детей правдивы. В противном случае выведите «NO».

### Примеры

стандартный ввод	стандартный вывод
3 2 1 0	YES
3 2 0 0	NO

### Замечание

Возможные исходы для первого примера: партия между первым и третьим ребёнком завершилась победой первого, партия между первым и вторым завершилась победой первого, партия между вторым и третьим завершилась победой второго.

## Задача Н. Чудесная находка

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

Недавно Ямочка нашла чудесный титановый шар и решила похвастаться им перед всеми в городе. Всего в городе проживает  $N$  человек, каждый из которых имеет свой уникальный номер от 1 до  $N$ , причем Ямочка имеет номер 1.

Ввиду особенностей молекулярной структуры шара передавать его можно только внутри города и только по необычной *миндальной* связи. Всего  $N - 1$  пар людей в городе имеют такую связь друг с другом, однако гарантируется, что от Ямочки шар по миндальной связи может добраться до любого человека, возможно, через посредников.

Когда кто-то получает шар, он записывает на нём свой номер, после чего отдаёт шар человеку, с которым у него имеется миндальная связь и который ещё не видел шар, если такой имеется, иначе он возвращает шар тому, от кого получил его впервые. Ямочка действует по такому же принципу с одним исключением: когда все, кому она могла передать шар, уже его видели, она оставляет шар у себя и больше ничего не делает.

Очевидно, что когда все посмотрели на шар, на нём осталась очень необычная и интересная последовательность чисел. Ямочку теперь интересует вопрос, сколько различных последовательностей может получиться после передачи шара между всеми в городе? Помогите Ямочке и посчитайте их количество. Две последовательности  $a$  и  $b$  считаются различными, если их длины различаются, либо существует индекс  $1 \leq i \leq |a|$ , такой что  $a_i \neq b_i$ .

### Формат входных данных

В первой строке дано одно целое число  $N$  — количество человек в городе Ямочки ( $1 \leq N \leq 10^5$ ).

В следующих  $N - 1$  строках даны по два целых числа  $x_i$  и  $y_i$  — пара людей, между которыми существует миндальная связь ( $1 \leq x_i, y_i \leq N$ ). Гарантируется, что  $x_i \neq y_i$ .

### Формат выходных данных

Выведите единственное число — количество различных последовательностей. Так как ответ может быть слишком большим, выведите его остаток от деления по модулю на  $10^9 + 7$ .

### Примеры

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

### Замечание

В первом примере существует всего одна последовательность —  $[1, 2, 3, 2, 1]$ .

Во втором примере есть 4 последовательности:

- $[1, 3, 1, 2, 4, 2, 5, 2, 1]$ ;
- $[1, 3, 1, 2, 5, 2, 4, 2, 1]$ ;
- $[1, 2, 4, 2, 5, 2, 1, 3, 1]$ ;
- $[1, 2, 5, 2, 4, 2, 1, 3, 1]$ .

## Задача I. Большой лифт

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

В 2084-м году была построена новая штаб-квартира компании из  $N$  этажей, этажи нумеруются от 1 до  $N$ . В здании построили  $K$  лифтов, каждый из которых ездит по своему отрезку этажей  $[l_i, r_i]$ , то есть  $i$ -й лифт может переместить человека между любыми двумя этажами из отрезка  $[l_i, r_i]$ . Перемещаться между этажами здания можно только используя лифты. Гарантируется, что существует способ добраться с любого этажа здания на любой другой.

Из-за этого возникла проблема: главный HR-специалист потребовал установить в некоторых лифтах камеры так, чтобы каждый человек, заходящий в здание на первый этаж и поднимающийся на верхний этаж  $N$ , был замечен хотя бы одной камерой. Камера замечает человека, только если он использует лифт, в котором эта камера установлена. Однако покупка каждой камеры требует денег, поэтому нужно разместить как можно меньше камер.

### Формат входных данных

Первая строка содержит два целых числа  $N$  и  $K$  — количество этажей и лифтов соответственно ( $1 \leq N, K \leq 10^5$ ).

Следующие  $K$  строк содержат по два целых числа  $l_i$  и  $r_i$  — границы отрезка этажей, по которому ездит  $i$ -й лифт ( $1 \leq l_i \leq r_i \leq N$ ).

### Формат выходных данных

Выведите одно число — минимальное количество камер, которое потребуется, чтобы выполнить требование HR-специалиста, или  $-1$ , если выполнить требование не возможно.

### Примеры

стандартный ввод	стандартный вывод
1984 1 1 1984	1
2 2 1 2 1 2	2

## Задача J. Двоичная сортировка

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

Недавно Тимофей начал изучать алгоритмы и сейчас проходит сортировку. Чтобы закрепить материал, Тимофей решил поупражняться на строке  $S$  длины  $N$ , состоящей из нулей и единиц. Отсортировать строку целиком было бы слишком просто, поэтому Тимофей решил сортировать подстроки  $S$ , то есть он умеет делать следующую операцию:

- выбрать два индекса  $l$  и  $r$  ( $1 \leq l \leq r \leq N$ ) и отсортировать подстроку  $S_l S_{l+1} \dots S_r$  в порядке неубывания.

Чтобы хорошо потренироваться, Тимофей поставил себе цель. Он хочет, применив сортировку подстроки неограниченное количество раз (возможно, ноль), получить из строки  $S$  палиндром. Скажите, может ли Тимофей добиться своей цели?

Палиндромом называется строка, которая читается одинаково как слева направо, так и справа налево.

### Формат входных данных

Первая строка содержит единственное целое число  $N$  — длину строки  $S$  ( $1 \leq N \leq 10^5$ ).

Вторая строка содержит строку  $S$ , состоящую только из символов 0 и 1.

### Формат выходных данных

Выведите «YES», если существует последовательность операций, превращающих строку  $S$  в палиндром, и «NO» в противном случае.

### Примеры

стандартный ввод	стандартный вывод
12 010110110100	YES
1 1	YES
2 10	NO

### Замечание

В первом примере можно получить строку 010011110010:

- сначала отсортировать  $l = 4, r = 6$ , получая 010011110100.
- затем  $l = 10, r = 11$ , получая 010011110010.

## Задача К. Пробег

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

Недавно Вадим придумал новый вид спорта «Пробег»: совмещение программирования и бега. Если сильно упростить правила, цель пробега — пробежать конкретное расстояние до компьютера и решить на нём задачу. Известно, что каждый из  $N$  компьютеров расположен на декартовой плоскости, причём каждый из них находится в своей точке.

Авторы сегодняшнего конкурса решили сыграть в пробег. Известно, что всего их  $Q$  человек, причём каждый стоит где-то на плоскости (необязательно в различных точках), и у каждого есть своя величина  $d_j$  — расстояние, которое нужно пробежать  $j$ -му человеку до одного из компьютеров. При этом авторы решили, что бежать можно только параллельно осям координат, причём если до какого-то компьютера возможно добраться за меньшее расстояние, чем требуется от человека, то до этого компьютера бежать нельзя.

Изобретатель пробега и автор сегодняшнего конкурса Вадим заранее знает, где будут находиться компьютеры, где будут стоять все участники пробега и требуемое расстояние для каждого. Чтобы иметь небольшое преимущество над конкурентами, Вадим захотел посчитать, сколько различных компьютеров могут подойти каждому из участников пробега. Помогите ему найти эти количества.

### Формат входных данных

В первой строке дано целое число  $N$  — количество компьютеров ( $1 \leq N \leq 10^5$ ).

В следующих  $N$  строках даны два целых числа  $cx_i$  и  $cy_i$  — координаты  $i$ -го компьютера ( $|cx_i|, |cy_i| \leq 10^9$ ).

Гарантируется, что никакие два компьютера не стоят в одной точке.

В следующей строке дано целое число  $Q$  — количество участников пробега ( $1 \leq Q \leq 10^5$ ).

В следующих  $Q$  строках даны три целых числа  $qx_j$ ,  $qy_j$  и  $d_j$  — координаты  $j$ -го участника и требуемое от него расстояние для бега ( $|qx_j|, |qy_j| \leq 10^9, 0 \leq d_j \leq 4 \cdot 10^9$ ).

### Формат выходных данных

Выведите  $Q$  целых чисел  $m_j$  — количество компьютеров, до которых может добежать  $j$ -й участник пробега.

### Пример

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

## Задача L. Философский вопрос

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

В новом семестре Ваня решил серьёзно взяться за учебу и перестать прогуливать пары, поэтому сейчас он торопится на пару по философии. Как известно, любое здание университета представляет собой систему из  $N$  аудиторий, пронумерованных натуральными числами от 1 до  $N$ , и  $M$  коридоров между ними. Ванин университет не исключение. Любому студенту известно, что все коридоры имеют одинаковую длину, по любому из коридоров можно ходить в любую сторону, никакой коридор не соединяет аудиторию саму с собой, никакие две аудитории не соединены больше, чем одним коридором, и что из любой аудитории можно добраться в любую, не покидая здание.

Поднявшись на шестой этаж, Ваня понял, что больше не может сделать ни шагу, не выпив перед этим кофе. К счастью, у Вани с собой есть целых  $K$  стаканчиков кофе, он точно знает, что, выпив  $i$ -й стакан, сможет пройти ровно  $a_i$  коридоров (даже если он дойдет до нужной аудитории, посетив меньше коридоров, то всё равно не сможет остановиться, пока не пройдёт  $a_i$  коридоров).

Сейчас Ваня находится в аудитории с номером 1 и ему нужно решить философский вопрос — сколько же стаканчиков кофе нужно выпить, чтобы дойти до нужной аудитории. Ваня опаздывает, поэтому он хочет выпить наименьшее число. Ещё одна проблема в том, что он точно не помнит, в какой аудитории будет проходить пара, поэтому хочет ответить на вопрос для каждой возможной аудитории независимо.

Более формально, для каждой аудитории  $v$  независимо надо найти такое наименьшее число  $x$ , что существует такое подмножество стаканчиков с индексами  $i_1, i_2, \dots, i_x$ , для которого найдётся путь из аудитории 1 в аудиторию  $v$ , состоящий ровно из  $a_{i_1} + a_{i_2} + \dots + a_{i_x}$  коридоров. Аудитории и коридоры в пути могут использоваться более одного раза, в частности, можно пройти по какому-то коридору из аудитории  $a$  в  $b$  и затем сразу же вернуться снова в  $a$ . Если такого  $x$  не существует, то выведите вместо него  $-1$ . Сумму пустого подмножества стаканчиков можно считать равной 0.

### Формат входных данных

В первой строке содержатся три целых числа  $N$ ,  $M$  и  $K$  — количество аудиторий, коридоров и стаканчиков кофе соответственно ( $2 \leq N \leq 10^5$ ;  $N - 1 \leq M \leq 2 \cdot 10^5$ ;  $1 \leq K \leq 10^5$ ).

Следующие  $M$  строк содержат по два целых числа  $u_i$  и  $v_i$  — описание очередного коридора, соединяющего аудитории  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq N$ ). Гарантируется, что никакой коридор не соединяет аудиторию саму с собой, что каждый коридор во входных данных встречается не более одного раза и что предоставленный план университета является связным.

Следующая строка содержит  $K$  целых чисел  $a_1, a_2, \dots, a_K$  — описания стаканчиков кофе Вани ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

В единственной строке выведите  $N$  чисел — ответ для каждой вершины.

## Примеры

стандартный ввод	стандартный вывод
5 4 4 1 2 2 3 3 4 4 5 1 2 3 4	0 1 1 1 1
5 4 2 1 2 2 3 3 4 4 5 1 2	0 1 1 2 -1
2 1 1 1 2 999	0 1

## Замечание

В первом примере возможные расстояния до аудиторий 2, 3, 4, 5 равны 1, 2, 3, 4 коридоров соответственно. Для любого из этих расстояний найдётся один подходящий стаканчик.

Во втором примере до аудитории 5 можно добраться через 4 коридора, а кофе хватит максимум на  $2 + 1 = 3$  коридора.

В третьем примере, пройдя 999 раз по единственному коридору, Ваня в итоге окажется в аудитории 2.

## Задача М. Обратный к НОДу

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 512 мегабайт

Многие слышали и использовали операцию нахождения наибольшего общего делителя. Обычно эта операция записывается как  $(n, m)$ . Но в этот раз Вадим решил рассмотреть числа, обратные к НОДу, то есть числа, равные  $\frac{1}{(n,m)}$ .

Эти числа очень заинтриговали Вадима, и он решил посчитать обратные к НОДу для всех пар чисел от 1 до  $N$ . К сожалению, это очень долгое занятие, поэтому Вадим просит вас помочь найти среднее арифметическое всех этих значений.

### Формат входных данных

В единственной строке вводится целое число  $N$  ( $1 \leq N \leq 10^7$ ).

### Формат выходных данных

Выведите ответ на задачу по модулю  $10^9 + 7$ . То есть если ваш ответ представляет из себя несократимую дробь  $\frac{p}{q}$ , то выведите такое целое  $x$ , что  $0 \leq x \leq 10^9 + 6$  и  $p \equiv q \cdot x \pmod{10^9 + 7}$ .

### Примеры

стандартный ввод	стандартный вывод
2	833333340
3	805555562
1	1

### Замечание

В первом примере есть три НОДа:  $(1, 1) = 1$ ,  $(1, 2) = 1$ ,  $(2, 2) = 2$ . Среднее арифметическое обратных к этим НОДам равно:

$$\frac{\frac{1}{1} + \frac{1}{1} + \frac{1}{2}}{3} = \frac{5}{6}$$

$$6 \cdot 833333340 = 5000000040 = 5 \cdot (10^9 + 7) + 5 \equiv 5 \pmod{10^9 + 7}$$

## Задача N. Convex Hull Treap

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

Совсем недавно Вадим кинул  $N$  точек на плоскость с декартовыми координатами. Все точки разлетелись повсюду. Поэтому Вадим решил навести порядок.

Вначале он взял самую верхнюю точку (точку с наибольшей ординатой) и назвал её *делегатом* всех точек. После чего он построил выпуклую оболочку на всех  $N$  точках. Затем он выбрал все точки левее делегата (точки с меньшей абсциссой) и выделил их в отдельное множество. То же самое он сделал со всеми точками правее делегата (точки с большей абсциссой). С этими множествами точек и со всеми последующими Вадим проделал аналогичные операции. Если какое-либо множество оказывалось пустым, то Вадим пропускал его.

В итоге каждая точка побывала делегатом какого-то множества точек, на каждом из таких множеств Вадим построил выпуклую оболочку. В силу свойств выпуклой оболочки каждая из них является либо точкой, либо отрезком, либо выпуклым многоугольником. Назовём *размером* выпуклой оболочки для точки 1, для отрезка 2, а для выпуклого многоугольника — количество вершин этого многоугольника.

Найдите для каждой данной точки размер выпуклой оболочки множества, делегатом которого она является.

### Формат входных данных

В первой строке дано целое число  $N$  — количество точек, кинутых Вадимом ( $1 \leq N \leq 10^5$ ).

В следующих  $N$  строках даны два целых числа  $x_i$  и  $y_i$  — абсцисса и ордината  $i$ -й точки ( $|x_i|, |y_i| \leq 10^9$ ). Гарантируется, что все абсциссы различные и все ординаты различные.

### Формат выходных данных

Выведите  $N$  целых чисел  $l_i$  — размер выпуклой оболочки множества, делегатом которого является  $i$ -я точка.

### Пример

стандартный ввод	стандартный вывод
5	1
0 0	1
6 4	3
4 7	4
8 9	1
12 1	

## Задача O. Cavalry Battle Advanced

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

Недавно Вадим написал свою игру, которую он назвал «Cavalry Battle Advanced». Эта игра отдалённо напоминает шахматы, но отличается некоторыми важными моментами:

- В «Cavalry Battle Advanced» есть только одна фигура — шахматный конь. Эта фигура ходит буквой «Г» во всех направлениях так, что сначала двигается на 2 клетки по одной оси и на 1 — по другой. Если на клетке, на которую конь может попасть за один ход, стоит другая фигура, то конь бьёт эту фигуру.
- Изначальная расстановка фигур с стороны одного игрока происходит на поле размера  $3 \times N$ . При этом поставить можно сколько угодно коней, но только в пределах этого поля, игрок сам выбирает нужное ему количество.

Чтобы все игроки не заполняли всё доступное поле  $3 \times N$  конями, Вадим закодировал секретный бонус. Чтобы подробнее разобраться в нём, представим себе граф, в котором каждый поставленный конь является вершиной, а если пара коней бьют друг друга, то между соответствующими вершинами в графе стоит ребро. Бонус добавляется только в том случае, если этот граф является деревом (то есть этот граф связный и без циклов).

Предельно понятно, что все профессиональные игроки в «Cavalry Battle Advanced» хотят как получить этот секретный бонус, так и расставить при этом наибольшее возможное количество коней в обозначенное поле. Побудьте в роли этих игроков и найдите подобную расстановку.

### Формат входных данных

В единственной строке вводится целое число  $N$  — размер поля по горизонтали ( $1 \leq N \leq 10^5$ ).

### Формат выходных данных

Выведите расстановку в следующем формате: 3 строки по  $N$  чисел 0 или 1 — 0, если клетка пустая, и 1, если в клетке стоит конь. Таким образом, у вас получится таблица размера  $3 \times N$ . Все кони должны образовывать дерево описанным в условии образом.

Если есть несколько подходящих расстановок с наибольшим количеством коней, выведите любую из них.

### Примеры

стандартный ввод	стандартный вывод
3	0 1 1 1 0 1 1 1 1
2	1 0 0 0 0 1