

Задача А. Подъём

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

На что только не пойдут преподаватели ради того, чтобы вовремя разбудить детей и отправить их на линейку. Некоторые из них даже готовы показывать фокусы, например, вот такой.

В фокусе задействованы два человека, назовём их Семён и Егор. Изначально Егор отворачивается, а дети дают Семёну строку из 10 цифр. Семён заменяет одну из них на символ «?». После этого поворачивается Егор и (о чудо!) угадывает, какое число было спрятано.

Проблема заключается в том, что Семён и Егор пока не решили, как им договориться, чтобы фокус получился. Помогите им, ибо подъём уже близко.

Протокол взаимодействия

Это задача с двойным запуском. Ваше решение будет запущено дважды, один раз вы будете действовать за Семёна, второй раз — за Егора.

На первом запуске в первой строке находится строка «hide», во второй строке — строка из 10 цифр. Вам необходимо вывести ту же строку, заменив один из символов на «?».

На втором запуске в первой строке находится строка «guess», во второй строке — вывод вашей программы после первого запуска. Вам необходимо вывести одну цифру, которая была спрятана на первом запуске.

Примеры

стандартный ввод	стандартный вывод
hide 1423301094	14233010?4
guess 14233010?4	1423301094

Задача В. Перестановка

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

Мальчику Васе на день рождения в ЛКШ подарили перестановку P ! Естественно, он сразу же начал с ней экспериментировать. Прямо сейчас он хочет её отсортировать. Для сортировки Вася собирается несколько раз поменять местами два элемента перестановки. Просто так сортировать ему неинтересно, поэтому он выбрал число K и решил добавить дополнительные ограничения:

1. Осуществлять обмен между индексами i и j ($i < j$) можно только в случае, если $K \leq j - i$;
2. Осуществлять обмен между индексами i и j ($i < j$) можно только в случае, если $P_i > P_j$;
3. Для любой пары индексов (i, j) обмен между ними можно совершать не более одного раза.

Васе так нравится перестановка, что он хочет *максимизировать* количество операций, которые ему потребуются для упорядочивания перестановки.

Формат входных данных

В первой строке содержатся 2 целых числа N и K ($2 \leq N \leq 500$; $1 \leq K \leq N - 1$).

В следующей строке даются N целых чисел — элементы перестановки (каждый — от 1 до 10 включительно).

Формат выходных данных

В первой строке выведите одно число M — максимальное количество обменов, которого можно достичь.

В каждой из следующих M строк выведите позиции i, j индексов, на которых нужно обменять элементы местами (индексы нумеруются с 1), причём $i < j$.

Примеры

стандартный ввод	стандартный вывод
3 1 3 2 1	3 2 3 1 3 1 2
5 4 1 4 3 2 5	0
4 2 4 1 2 3	2 1 4 1 3
10 5 8 7 6 10 9 3 1 5 2 4	15 3 8 3 10 3 9 2 8 2 10 2 9 2 7 1 8 1 10 1 9 1 7 1 6 5 10 4 10 4 9

Задача С. Ремонт первого комповника

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

В первом комповнике спустя столько лет решили поменять плитку на полу, который можно представить как прямоугольник n на m . Заказали 4 разных вида плитки, которые можно пронумеровать от 1 до 4. 1-й вид — самый яркий, а 4-й самый тусклый. Плиток оказалось очень много, и у дизайнеров есть простор для творчества. Вначале они вместе поняли, что виды соседних по стороне клеток должны различаться ровно на 1, так как одинаковые это скучно, а большое различие сильно бросается в глаза. Как дальше уложить плитки, они не придумали, поэтому решили посмотреть, сколько вариантов есть с таким условием и уже потом подумать, сколько еще условий могут добавить.

Формат входных данных

В первой строке вводятся два целых числа n, m — размеры пола ($1 \leq n, m \leq 24$).

Формат выходных данных

Выведите количество способов выложить плитку на полу по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
1 2	6
3 2	34
7 10	657297226

Задача D. Садоводство

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

Вам надоело заниматься олимпиадным программированием, поэтому было решено посвятить оставшуюся часть жизни работе в саду.

Есть n яблонь, пронумерованных от 1 до n . Изначально (день 0) на деревьях ничего нет. Каждый день на i -й яблоне вырастает i яблок.

Вы q раз хотите собрать урожай — в d_i -й день будут собраны все яблоки с деревьев на отрезке $[l_i, r_i]$.

Вам стало интересно, а сколько денег вы сможете заработать на продаже фруктов. Для этого после каждого сбора урожая нужно знать, сколько яблок вы собрали.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 10^{18}$) — количество яблонь.

На второй строке находится число q ($1 \leq q \leq 2 \cdot 10^5$) — количество раз, которое вы соберете урожай.

Следующие q строк описывают сборы урожая. В i -й из них даны числа d_i, l_i, r_i ($1 \leq l_i \leq r_i \leq n$, $1 \leq d_i \leq 10^{18}$).

Гарантируется, что последовательность d_i возрастает.

Формат выходных данных

Выведите q строк. В i -й из них сообщите, сколько яблок будет собрано во время i -го сбора урожая по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
5 3	10
2 2 3	30
3 3 5	35
5 1 5	
711741968610511029 1	287488159
82803127126515475 516874490286751784 588060532192410838	

Задача Е. Остров сокровищ

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

У вас есть карта сокровищ, на которой отмечено n пиратских сундуков, закопанных в землю и пронумерованных от 1 до n . i -й сундук находится в точке (i, p_i) на плоскости. Когда вы откроете i -й сундук, вы получите v_i монет.

Вы начинаете путь в точке $(0, 0)$. Каждый раз вы можете переместиться либо из (x, y) в $(x+1, y)$, либо в $(x, y+1)$. Вы можете выкопать сундук и забрать из него все монеты, только если находитесь прямо над ним.

К сожалению, на карте не отмечена опасная ловушка! После долгих исследований вы узнали, что это — прямоугольная область со сторонами, параллельными осям координат, и что существует ровно m возможных вариантов её расположения.

Напишите программу, которая для каждого варианта положения ловушки вычислит максимальное количество монет, которое можно собрать, следуя оптимальному пути.

Если реальная ловушка — это i -й прямоугольник $[x_1, x_2] \times [y_1, y_2]$, то точка (x, y) считается опасной тогда и только тогда, когда $x_1 \leq x \leq x_2$ и $y_1 \leq y \leq y_2$. Через опасные точки проходить нельзя.

Формат входных данных

Первая строка содержит одно целое число T ($1 \leq T \leq 100$) — количество наборов тестовых данных.

Далее следует описание наборов входных данных.

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 300\,000$) — количество сундуков и количество возможных ловушек.

Следующие n строк содержат по два целых числа p_i и v_i ($1 \leq p_i \leq n$, $1 \leq v_i \leq 10^9$) — координата y сундука и количество монет в нём. Значения всех p_i попарно различны.

Следующие m строк содержат по четыре целых числа x_1, y_1, x_2, y_2 ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq n$) — описание возможной прямоугольной ловушки.

Гарантируется, что сумма всех n по всем тестам не превышает 1 500 000, и сумма всех m по всем тестам также не превышает 1 500 000. (Это неправда, немного меньше, но я не знаю сколько).

Формат выходных данных

Для каждого набора тестов выведите m строк: i -я строка должна содержать одно целое число — максимальное количество монет, которое можно собрать, если ловушка — i -й прямоугольник.

Пример

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

Задача F. Кто такой кейл

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

Мистер Нежмах начал выращивать кейл. Он хочет вырастить всего n головок кейла за следующие $2n$ дней. В i -й день Нежмах может:

- посадить одну головку кейла с вкусностью a_i ;
- собрать одну головку кейла.

Каждый раз, когда он решает собрать кейл, должен существовать хотя бы один кейл, который был посажен ранее и ещё не был собран.

Общая вкусность равна сумме вкусностей всех собранных головок кейла. Помогите Нежмаху составить план, который максимизирует суммарную вкусность!

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 2 \cdot 10^4$) — количество наборов входных данных.

Далее следует описание входных данных.

Первая строка содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$).

Вторая строка содержит массив a ($1 \leq a_i \leq 10^9$), где a_i — вкусность кейла, который можно посадить в i -й день.

Гарантируется, что сумма всех n по всем тестам не превышает $2 \cdot 10^5$.

Формат выходных данных

Для каждого теста выведите n чисел — дни, в которые Нежмах должен сажать кейл. Если существует несколько решений, можно вывести любое.

Пример

стандартный ввод	стандартный вывод
3	1
1	1 3 4
4 3	1 3
3	
1 2 4 9 6 3	
2	
5 100 100 200	