

Задача А. Построить башни

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

У Вадима есть N рублей, эти деньги он хочет потратить на постройку ханойских башен. Для одной такой ему потребуется одно основание, которое стоит B рублей, а также не менее, чем K колец, каждое из которых стоит R рублей. Помогите Вадиму ответить на два вопроса: какое наибольшее количество башен он сможет построить и, если сможет построить хотя бы одну, какая будет наибольшая возможная высота башни (количество колец) при найденном количестве?

Формат входных данных

В первой строке дано целое число N — сколько рублей есть у Вадима ($1 \leq N \leq 10^9$).

Во второй строке дано целое число B — стоимость одного основания ($1 \leq B \leq 10^9$).

В третьей строке дано целое число R — стоимость одного кольца ($1 \leq R \leq 10^9$).

В четвёртой строке дано целое число K — наименьшее количество колец, необходимых для постройки одной ханойской башне ($1 \leq K \leq 10^9$).

Формат выходных данных

Выведите 0, если Вадим не сможет построить ни одной башни, в противном случае выведите наибольшее количество башен, которое он сможет построить, и наибольшую возможную высоту при таком количестве башен.

Примеры

стандартный ввод	стандартный вывод
20 1 2 4	2 5
10 4 3 5	0

Замечание

В первом примере Вадим может потратить все деньги на постройку двух башен высотами 5 и 4.

Задача В. Пододеяльник

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

Вадим решил сшить себе новый пододеяльник. Для этого ему потребуются две одинаковые простыни, которые он сошьёт между собой по границе. Понятно, что чем больше периметр у простыней, тем больше будет потрачено ниток. Вадим обожает выпуклые многоугольники, поэтому хочет, чтобы его пододеяльник был такой формы. Также Вадиму не нужен слишком большой пододеяльник, потому что кровать у него не настолько большая. Формально говоря, абсциссы вершин многоугольника по модулю не должны превосходить X , и ординаты вершин многоугольника по модулю не должны превосходить Y .

Вадим хочет попрактиковаться в технологии шитья, поэтому хочет потратить самое большое возможное количество ниток. Помогите Вадиму найти форму простыней, соответствующую его пожеланиям.

Формат входных данных

В первой строке дано целое число X ($1 \leq X \leq 10^9$).
Во второй строке дано целое число Y ($1 \leq Y \leq 10^9$).

Формат выходных данных

Выведите нужную форму простыней в следующем формате: вначале выведите N — количество вершин искомого многоугольника ($3 \leq N \leq 10^5$). Затем выведите N пар целых чисел x_i и y_i — абсцисса и ордината i -й вершины многоугольника в порядке обхода ($|x_i| \leq X, |y_i| \leq Y$).

Ваш многоугольник должен быть без самопересечений, а также должен быть выпуклым.

Пример

стандартный ввод	стандартный вывод
1	3
1	1 1
	1 -1
	-1 0

Замечание

Обратите внимание, что ответ на данный в условии пример не является правильным. Он дан для демонстрации формата выходных данных.

Задача С. Завали программу Егора / Опять переполнение / Очень длинный int

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

Егор опять получил *WA* из-за переполнения переменной и решил, что больше так продолжаться не может. Для того, чтобы в будущем избежать таких досадных ошибок, Егор создал свой беззнаковый целочисленный тип, занимающий N бит памяти. В задаче, которую он решал, необходимо A раз вывести числа, такие что при прибавлении к ним T не получается целая неотрицательная степень двойки. Егор настолько уверен в том, что его целочисленный тип не переполнится, что каждый раз умножает предыдущее выведенное число на 2, прибавляет к нему T (то есть преобразует k_i в $2 \cdot k_i + T$) и выводит результат на экран. В качестве первого числа Егор выводит K .

Однако злым составителям тестов это не понравилось, и они решили придумать тест, на котором Егор обязательно получит *WA*. Такой тест должен состоять из 1 натурального числа A — количества чисел, которое необходимо вывести программе, чтобы хоть одно из чисел было ровно на T меньше степени двойки. Вам как главному составителю задач поручили найти минимальное такое A .

Формат входных данных

В первой и единственной строке вводятся 3 целых числа: N ($1 \leq N \leq 10^9$), K ($1 - T \leq K \leq \min(10^9, 2^N - 1)$), T ($0 \leq T \leq 1$) — длина типа Егора; первое число, выводимое его программой и число T из задачи, которую он решал.

Формат выходных данных

Выведите единственное число A — минимальный тест, который не пройдет программа Егора.

Примеры

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

Замечание

В первом примере число K уже на 1 меньше степени двойки, а значит, решение Егора получит *WA* даже при $A = 1$.

Во втором примере $7 = 0111_2$, а значит, чтобы оно стало степенью двойки из-за переполнения, его нужно умножить на $8 = 2^3$.

Задача D. Сергей и базы данных

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

У Семёна есть база данных, в которой можно выполнять те или иные запросы. Как-то раз Сергей забрёл к Семёну и увидел одну из баз данных, в которой было два столбца A и B , которые содержат целые числа a_i и b_i соответственно. Сергей только начал изучать базы данных, и единственное, что он умеет на этот момент — это выполнять всего лишь один SQL-запрос:

```
SELECT sum(B) WHERE A =  $x_1$  OR A =  $x_1$  OR ... OR A =  $x_K$ 
```

Недолго поразмышляв, Сергей решил применить знания на практике и найти K таких целых чисел, что при подстановке их в данный запрос ответ-сумма была максимальной.

Семён уже знает, что делает этот запрос, поэтому решил проверить Сергея, понимает ли он. Сергей очень быстро ответил, что этот запрос берёт сумму всех b_i , напротив которых стоит a_i , равное одному из x_j в его запросе. Если подходящих под запрос a_i нет, то ответ на запрос 0.

Сергей так увлечён изучением этой базы данных и новых SQL-запросов, что попросил Вас найти K таких x_j , чтобы ответ-сумма на данный запрос была максимальной.

Формат входных данных

Первая строка входных данных содержит целые числа N и K , такие что $1 \leq N, K \leq 10^5$.

В следующих N строках даны пары целых чисел a_i и b_i , такие что $0 \leq a_i \leq 10^5$ и $-10^9 \leq b_i \leq 10^9$.

Формат выходных данных

Выведите K целых чисел x_i , при которых запрос в базу данных будет максимален ($-10^9 \leq x_i \leq 10^9$). Если подходящих ответов несколько, выведите любой из них.

Примеры

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

Задача Е. Марго делает домашнее задание

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

Сегодня ученица первого класса Марго решила, что обязательно сделает домашку по информатике. Ей дан массив длины N , состоящий из целых чисел, и Q вопросов следующего вида:

- ! X : посчитать, сколько в массиве чисел меньше X , сколько чисел равно X и сколько чисел больше X ;
- + X : увеличить все элементы массива на X .

К сожалению, для Марго эта задача слишком сложная (это простительно, ведь она только в первом классе), поэтому ей требуется Ваша помощь.

Формат входных данных

В первой строке дано целое число N ($1 \leq N \leq 1.5 \cdot 10^5$) — размер массива.

Во второй строке даны N целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — исходный массив.

В третьей строке дано целое число Q ($1 \leq Q \leq 1.5 \cdot 10^5$) — количество вопросов в домашнем задании.

Каждая из последующих Q строк содержит символ T и число X ($-10^9 \leq X \leq 10^9$) — описание очередного вопроса. Гарантируется, что каждый вопрос является корректным.

Формат выходных данных

Для каждого вопроса первого типа выведите в одной строке три числа — ответ на очередной вопрос.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5 5 ! -1 ! 1 ! 3 ! 5 ! 7	0 0 5 0 1 4 2 1 2 4 1 0 5 0 0
4 1 3 3 7 3 ! 4 + 1 ! 4	3 0 1 1 2 1

Задача F. Сбалансированная олимпиада

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

Для одного соревнования по олимпиадной информатике жюри подготовили N задач, i -я из которых обладает сложностью A_i .

Незадолго до начала олимпиады составители поняли, что текущий набор задач является слишком несбалансированным — некоторые сложности задач встречались в наборе гораздо чаще других. Поскольку времени, чтобы исправить свою ошибку, у жюри почти не осталось, они решили использовать в соревновании несколько задач, последовательно идущих в изначальном контексте.

Разумеется, необходимо, чтобы выбранный набор задач был сбалансированным — каждая уникальная сложность в нем должна встречаться одинаковое количество раз. Например, наборы $[2, 3, 2, 3]$, $[1]$, $[5, 5, 100, 121, 100, 121, 5, 121, 100]$ являются таковыми, а наборы $[1, 2, 2]$, $[2, 2, 3, 3, 4, 4, 1]$ — нет.

Поскольку составителям очень нравятся приготовленные задачи на эту олимпиаду, то среди всех подходящих наборов они хотят выбрать самый длинный — тогда участники точно не заскучают на соревновании.

Помогите жюри найти, какие задачи нужно оставить в контексте, чтобы он стал сбалансированным.

Формат входных данных

В первой строке вводится количество изначально приготовленных задач N ($1 \leq N \leq 5000$).

Во второй строке вводятся N целых чисел A_i ($1 \leq A_i \leq N$) — сложности задач в олимпиаде.

Формат выходных данных

В единственной строке нужно вывести два целых числа — номера первой и последней задач набора, взятого на соревнование. Если подходящих наборов несколько, выведите любой из них.

Примеры

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

Задача G. Сломанная программа 2

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

Робот находится в лабиринте. Лабиринт представлен в виде таблицы $N \times M$ символов. '#' на пересечении i -й строки и j -го столбца в таблице обозначает, что на координатах (i, j) в лабиринте находится стена, а '.' — её отсутствие. Робот находится на координатах (x, y) . Гарантируется, что на этих координатах нет стены, а также что из лабиринта нет выхода. Изначально робот имеет прочность C .

Роботу дана программа из K команд. Команды задаются в виде « $u s$ », где u — символ, обозначающий направление движения ('U' — вверх, 'D' — вниз, 'R' — вправо, 'L' — влево), а s — целое число, обозначающее, сколько ячеек робот пройдет в указанном направлении. При этом может случиться такое, что при исполнении очередной команды робот «упрется в стенку» — то есть следующая клетка его движения окажется стеной. В таком случае он продолжит исполнять текущую команду, но вместо движения будет терять по одной единице прочности за каждую оставшуюся ячейку пути.

Если робот сможет выполнить все команды, выведите число 0, иначе выведите номер команды, при выполнении которой он сломается.

Формат входных данных

В первой строке находятся 2 числа N и M ($3 \leq N, M \leq 200\,000, N \times M \leq 200\,000$).

В следующих N строках находятся M символов '.' и '#' — описание лабиринта.

В следующей строке находятся 3 числа x, y , и C ($1 < x < N, 1 < y < M, 1 \leq C \leq 10^9$) — координаты, на которых изначально стоит робот и его прочность соответственно.

В следующей строке находится число K ($1 \leq K \leq 100\,000$) — количество команд в программе робота.

В следующих K строках находятся символ u и число s ($1 \leq s \leq 100\,000$) — описание команд.

Гарантируется, что все граничные клетки лабиринта являются стенами. Граничными клетками называются клетки первой и N -й строк, а также первого и M -го столбцов.

Формат выходных данных

Выведите единственное число — номер команды, при выполнении которой робот сломается, или число 0, если он сможет выполнить все команды.

Пример

стандартный ввод	стандартный вывод
5 6 ##### #...# #.#.# #...## ##### 3 4 5 6 U 4 L 3 D 3 R 2 U 1 R 2	3

Задача Н. Миграция великанов

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

Великаны — очень интересные существа, они каждый год меняют своё место обитания. Так, на планете есть N мест обитаний великанов, а также M рейсов, связывающих два различных места обитания. Каждый рейс для великана стоит определённое количество денег, поэтому, находясь в одном месте обитания, они выбирают своим следующим такое, что выполняются следующие условия:

1. До него можно долететь ровно одним рейсом;
2. До него добираться дешевле всего;
3. Он в нём ещё не был.

Если великан не может выбрать ни одного следующего места обитания, то он остаётся на его текущем навсегда. В каждом из мест обитаний стартует по одному великану. Ответьте, в каком месте обитания каждый из них останется навсегда.

Формат входных данных

В первой строке даны два целых числа N и M — количество мест обитаний на планете и количество рейсов ($2 \leq N \leq 10^5, 1 \leq M \leq 10^5$).

В следующих M строках описаны рейсы тремя целыми числами u_i, v_i и c_i — два места обитания, которые связывает i -й рейс, а также его стоимость ($1 \leq u_i, v_i \leq N, u_i \neq v_i, 1 \leq c_i \leq 10^9$).

Гарантируется, что никакие два рейса не связывают одну и ту же пару мест обитаний. Гарантируется, что все стоимости рейсов различные.

Формат выходных данных

Выведите N целых чисел f_i — индекс места обитания, в котором в конце концов окажется великан, начинающий в месте обитания с индексом i .

Пример

стандартный ввод	стандартный вывод
5 4	4
1 5 15	4
2 5 14	3
2 4 10	2
5 4 18	4

Замечание

Великан, стартующий в третьем месте обитания, останется в нём же, потому что оттуда не следует ни одного рейса.

Великан, стартующий в четвёртом месте обитания, в первый год переместится во второе, а во второй год он захочет переместиться в четвёртое (потому что до него добираться дешевле всего), но не сделает этого, так как уже был там, из-за этого он останется во втором навсегда.